Titel der Arbeit:
Efficient Secure Communication in VANETs under the Presence of new Requirements Emerging from Advanced Attacks

Dissertation
zur Erlangung des akademischen Grades

Doktor-Ingenieur

(Dr.-Ing.)

im Fach: Informatik

eingereicht an der

Mathematisch-Naturwissenschaftlichen Fakultät

der Humboldt-Universität zu Berlin

von

Dipl.-Ing. (FH) Sebastian Bittl, M.Sc. (hons)

Präsidentin der Humboldt-Universität zu Berlin
Prof. Dr.-Ing. Dr. Sabine Kunst

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät
Prof. Dr. rer. nat. Elmar Kulke

Gutachter:	1. Prof. Dr. rer. nat. Björn Scheuermann 2. Prof. Dr. rer. nat. Frank Kargl 3. Prof. Dr.-Ing. Falko Dressler
Tag der mündlichen Prüfung:	11.09.2017

Ich erkläre, dass ich die Dissertation selbständig und nur unter Verwendung der von mir gemäß § 7 Abs. 3 der Promotionsordnung der Mathematisch-Naturwissenschaftlichen Fakultät, veröffentlicht im Amtlichen Mitteilungsblatt der Humboldt-Universität zu Berlin Nr. 126/2014 am 18.11.2014 angegebenen Hilfsmittel angefertigt habe.

Acknowledgements

This thesis was started during my time as a research engineer at Fraunhofer ESK in Munich and continued at the Chair of Computer Engineering at Humboldt University of Berlin. I am very thankful to Professor Björn Scheuermann for his support and for the opportunity to finish this work under his supervision.

I also thank Professor Frank Kargl for his support and for acting as a co-referee. Moreover, I also thank Professor Falko Dressler for acting as a co-referee and Professor Ralf Reulke for acting as the chairman of the examination board.

I also want to thank my former colleagues at Fraunhofer ESK who contributed to a good working atmosphere and were always available for valuable discussions, especially Karsten Roscher and Arturo A. Gonzalez.

Finally, I am very thankful to my family for their great support.

Contents

List of Figures

List of Tables

List of Listings

During work on this thesis the following work was published:

- S. Bittl, “Attack Potential and Efficient Security Enhancement of Automotive Bus Networks using Short MACs with Rapid Key Change”, in *6th International Workshop Nets4Cars/Nets4Trains/Nets4Aircraft*, ser. LNCS, vol. 8435. Springer, May 2014, pp. 113 - 125

- S. Bittl, A. A. Gonzalez and W. Heidrich, “Performance Comparision of Encoding Schemes for ETSI ITS C2X Communication Systems”, in *Third International Conference on Advances in Vehicular Systems, Technologies and Applications*, ThinkMind(TM) Digital Library, June 2014, pp. 58 - 63

- S. Bittl and K. Roscher, “Towards Efficient Methodologies for Rapid-prototyping of Communication Technologies and Cooperative ITS Applications”, in *ICWMC 2014 / VEHICULAR 2014 Expert Panel: Mobility: Achievements and Challenges*, ThinkMind(TM) Digital Library, June 2014, pp. 29 - 34

- S. Bittl, “Efficient Construction of Infinite Length Hash Chains with Perfect Forward Secrecy using Two Independent Hash Functions”, in *11th International Conference on Security and Cryptography*, SCITEPRESS Digital Library, Aug. 2014, pp. 213 - 220

- S. Bittl, “Angriffspotentiale und Effiziente Absicherung Automobiler Bussysteme als Grundlage sicherer Car2X-Kommunikation”, in *11th Conference Wireless Communication and Information*, vwh, Oct. 2014, pp. 37 - 49

- K. Roscher, S. Bittl, A. A. Gonzalez, M. Myrtus and J. Jiru, “ezCar2X: Rapid-Prototyping of Communication Technologies and Cooperative ITS Applications on Real Targets and Inside Simulation Environments”, in *11th Conference Wireless Communication and Information*, vwh, Oct. 2014, pp. 51 - 62

- S. Bittl and A. A. Gonzalez, “Privacy Issues and Pitfalls in VANET Standards”, in *1st International Conference on Vehicular Intelligent Transport Systems*, SCITEPRESS Digital Library, May 2015, pp. 144 - 151

- S. Bittl, B. Aydinli and K. Roscher, “Effective Certificate Distribution in ETSI ITS VANETs using Implicit and Explicit Requests”, in *8th International Workshop Nets4Cars/Nets4Trains/Nets4Aircraft*, ser. LNCS, vol. 9066. Springer, May 2015, pp. 72 - 83

- S. Bittl, A. A. Gonzalez, M. Spähn and W. Heidrich, “Performance Comparison of Data Serialization Schemes for ETSI ITS Car-to-X Communication Systems”, in *International Journal On Advances in Telecommunications*, June 2015, pp. 48 - 58

- S. Bittl, B. Aydinli and K. Roscher, “Efficient Rate-Adaptive Certificate Distribution in VANETs”, in *Twelfth International Symposium on Wireless Communication Systems*, IEEE Xplore Digital Library, Aug. 2015, pp. 371 - 375

- S. Bittl, A. A. Gonzalez, M. Myrtus, H. Beckmann, S. Sailer and B. Eissfeller, "Emerging Attacks on VANET Security based on GPS Time Spoofing", in *IEEE Communications and Network Security Conference*, IEEE Xplore Digital Library, Sept. 2015, pp. 344 - 352

- S. Bittl, "Efficient Distribution of Static or Slowly Changing Configuration Parameters in VANETs", in *9th International Workshop Nets4Cars*, Oct. 2015, pp. 301 - 306

- S. Bittl, K. Roscher and A. A. Gonzalez, "Security Overhead and its Impact in VANETs", in *8th IFIP Wireless Mobile Networking Conference*, IEEE Xplore Digital Library, Oct. 2015, pp. 192 - 199

- S. Bittl, B. Aydinli and K. Roscher, "Distribution of Pseudonym Certificates via Bursts for VANETs with Low and Medium Mobility", in *8th IFIP Wireless Mobile Networking Conference*, IEEE Xplore Digital Library, Oct. 2015, pp. 227 - 230

- S. Bittl, M. Schlegel and K. Roscher, "Simulationsbasierte Evaluierung eines zeit- und ortsbasierten Pseudonym-Wechsel-Verfahrens für ETSI ITS - Dezentraler Ansatz zur Verbesserung der Privatsphäre von Fahrern", in *31. VDI/VW-Gemeinschaftstagung Automotive Security*, ser. VDI-Berichte, vol. 2263. VDI Verlag, Oct. 2015, pp. 137 - 146

- S. Bittl and A. A. Gonzalez, "Privacy Endangerment from Protocol Data Sets in VANETs and Countermeasures", in *Smart Cities, Green Technologies, and Intelligent Transport Systems*, ser. CCIS, vol. 579. Springer, Jan. 2016, pp. 304 - 321

- S. Bittl and K. Roscher, "Efficient Authorization Authority Certificate Distribution in VANETs", in *2nd International Conference on Information Systems Security and Privacy*, SCITEPRESS Digital Library, Feb. 2016, pp. 85 - 96

- S. Bittl and K. Roscher, "Feasibility of Verify-on-Demand in VANETs", in *4th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication*, Institut für Informatik Humboldt-Universität zu Berlin, Apr. 2016, pp. 10 - 13

- D. Seydel, S. Bittl, J. Pfeiffer, J. Jiru, H. Beckmann, K. Frankl and B. Eissfeller, "An Evaluation Methodology for VANET Applications combining Simulation and Multi-sensor Experiments", in *2nd International Conference on Vehicle Technology and Intelligent Transport Systems*, SCITEPRESS Digital Library, Apr. 2016, pp. 213 - 224

- S. Bittl, "Towards Solutions for Current Security Related Issues in ETSI ITS", in *10th International Workshop Nets4Cars/Nets4Trains/Nets4Aircraft*, ser. LNCS, vol. 9669. Springer, May 2016, pp. 136 - 148

- S. Bittl, "Distributed Cross Layer Duplicate Address Handling for Safety Critical VANET Communication", in *Vehicular Communications*, Oct. 2016, pp. 1 - 6

- S. Bittl, "Safeguarding of Data Exchange", European Patent EP 3 133 769 A1, Feb., 2017.

- S. Bittl and K. Roscher, “Efficient Distribution of Certificate Chains in VANETs”, in *Information Systems Security and Privacy*, ser. CCIS, vol. 691. Springer, Feb. 2017, pp. 86 - 107
- S. Bittl, “Privacy Conserving Low Volume Information Retrieval from Backbone Services in VANETs”, in *Vehicular Communications*, 2017, Feb. 2017, pp. 1 - 7
- S. Bittl and K. Roscher, “Protocol Modeling Accuracy in VANET Simulators”, *5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication*, Apr. 2017, pp. 13 - 16
- S. Bittl and K. Roscher, “Mutual Influence of Certificate Distribution and Pseudonym Change Strategies in Vehicular ad-hoc Networks”, *International Journal of Vehicle Information and Communication Systems*, vol. 3, no. 2, Sept. 2017, pp. 158 - 172
- S. Bittl, D. Seydel, J. Pfeiffer, J. Jiru, “Evaluation Methodology for Cooperative ADAS utilizing Simulation and Experiments”, in *SMARTGREENS/VEHTIS 2016 - Revised Selected Papers*, ser. CCIS. Springer, to appear

Abstract

Vehicular ad-hoc networks are of high interest in both research and practice. They promise to enable realization of future safety critical advanced driver assistance systems, which help to increase safety of driving. Security and privacy of participants of such networks are core points of concern, due to wireless data exchange (often called car-to-X communication), which enables attacks even with only low effort spent. Hence, a security system dedicated to car-to-X communication has been developed. However, this work shows that significant overhead is caused by the existing solution. Moreover, several weaknesses leading to advanced attacks compromising system security are found and evaluated. Based on these findings, a set of extra requirements for realization of security systems for vehicular ad-hoc networks is determined. Approaches for extending European and US systems being currently standardized are proposed and evaluated, which make such systems conform to the newly identified requirements. Additionally, cross-layer design weaknesses interfering with security functionality are identified and proposals to overcome them are provided.

Chapter 1

Introduction

Traffic safety is a major concern of both government bodies as well as vehicle manufacturers. A high number of people being affected by traffic accidents is a worldwide problem [247, 305]. This inspired the declaration of the United Nations Decade of Action for Road Safety 2011–2020. Multiple initiatives have emerged from that declaration. One of them aims to develop and deploy innovative Advanced Driver Assistence Systems (ADASs), to increase traffic safety, especially in Europe and the USA [130, 305].

Cooperative driving enabled via so-called Car-to-X (C2X) (or Vehicle-to-X (V2X)) communication is regarded as an enabler for a large set of future safety critical ADASs. The development of corresponding technology has significantly sped up during recent years. Important progress towards standardization of corresponding mechanisms is made within European Telecommunications Standards Institute (ETSI) Intelligent Transport Systems (ITS) and United States Wireless Access in Vehicular Environments (WAVE) frameworks [1,59,129,154]. Within Europe major effort is taken within the Car2Car Communication Consortium (C2C-CC) [1].

To enable cooperative driving, participants, i.e., mobile vehicles being equipped with an on-board unit (OBU) and static road side units (RSUs), form a VANET. This kind of ad-hoc network is characterized by high vehicle dynamics and limited communication range of individual participants.

Moreover, dedicated wireless channels have been reserved for future VANETs in the 5.9 GHz band [126, 156]. However, available bandwidth is highly limited, causing high importance of bandwidth saving techniques throughout the protocol stack [59]. Furthermore, the automotive domain is highly cost sensitive [61]. Hence, expensive solutions to the existing challenges in VANETs are to be challenged by manufacturers and should be avoided.

One of the core points of concern for VANETs is secure and privacy-preserving message exchange. Thereby, wireless data transmission and intended realization of safety-critical ADAS based on exchanged data sets pose tough requirements for the security mechanisms. This holds especially, as a large number of attacks on VANETs has been proposed during the last years. Hence, while the provided level of security and privacy needs to be high, overhead caused by corresponding algorithms has to be kept to a minimum [59, 279].

This work takes an in detail look on the overhead caused by securing communication within VANETs, to identify possibilities to limit the extend of existing overhead in order to reduce the

impact of such overhead on overall system performance. Thereby, we find that several sources of overhead exist, and interaction of such sources has only been partly considered in prior work. Especially, the influence of on-demand certificate distribution is extended to the case of disseminating a multi-level certificate hierarchy. Thereby, the need for advanced distribution schemes for Certificate Authority (CA) certificates is identified, as the standardized straight forward solution is found to cause massive overhead. Additionally, a comparison of platform independent data representation schemes for security related meta data is provided. It shows that more than 9% in data length can be saved by using data representation based on Efficient XML Interchange (EXI) [311], instead of the standardized binary data representation scheme. Moreover, cross influence from certificate distribution and certificate change mechanisms is found to significantly limit the bandwidth saving approaches of certificate dissemination strategies.

Additionally, the given cross-layer aware protocol analysis identifies two severe protocol design weaknesses within ETSI ITS. High variance of content sizes on multiple protocol layers leads to violations of maximum data size limitations present within the ITS-G5 access layer. Hence, a cross-layer coordination scheme is introduced to avoid such conflicts. Moreover, the state of the art meta data handling at network layer and within the security functionality is found to completely disable required end-to-end encrypted multi-hop communication. A protocol improvement is suggested, which overcomes the found weakness.

Furthermore, a set of novel attacks on wireless data exchange in VANETs is identified. Conducted evaluations show a high impact of the attacks, which range from Denial of Service (DOS) weaknesses to Sybil and de-pseudonymization attacks. High overhead from certificate and especially certificate chain dissemination enables an attacker to easily perform a DOS attack, with an affected area significantly exceeding the communication range of a single attacker. Attacks on the Global Navigation Satellite System (GNSS) input of nodes allow a broad set of attacks, which span from a DOS attack to Sybil and message injection attacks. These illustrate the need to extend security mechanisms for important GNSS data input for VANETs. Additionally, an in detail analysis of the privacy impact of data sets from all protocol layers shows that may of them are highly characteristic for their sender, which enables fingerprinting. This leads to tracking and de-pseudonymization attacks on nodes.

Countermeasures to the found problems are proposed, which extend the security and privacy approaches from prior work. Both analysis, overhead as well as attacks, lead to a set of extra requirements for securing VANETs. Extensions for the European ETSI ITS and US WAVE VANET approaches are proposed and evaluated. Thereby, it is shown that the proposed improvements can be used to make these systems fulfill the newly found requirements. Identified attacks are either disabled or their impact is significantly limited by the proposed improvements.

In summary, this work addresses the research question of how to provide secure and privacy preserving message exchange in VANETs, while keeping the overhead caused by corresponding mechanisms low?

The further outline is as follows. At first, Chapter 2 provides an overview about fundamentals of the VANET domain and a review of related work. General aspects of the evaluation methodology applied throughout this work are given in Chapter 3. Then, an overview of overhead sources related to security functionality within VANETs is given in Chapter 4. Some kinds of identified overhead can be used for advanced DOS attacks on VANETs, as shown in

Chapter 5. Moreover, additional requirements for secure VANET communication are derived from further advanced attacks on such systems. In doing so, attacks on satellite based data input as well as privacy limitations from cross layer data sets are considered. Afterwards, efficient Public Key Infrastructure (PKI) handling is discussed in Chapter 6. Finally, a conclusion about achieved results can be found in Chapter 7 alongside with possible topics of future work.

Chapter 2

Fundamentals and Related Work

This chapter provides an overview about VANET fundamentals alongside with a review of related work in the areas covered within this work. The discussion is separated into

- basics of VANET approaches looked at in Section 2.1,
- VANET security architectures studied in Section 2.2,
- attacks on VANETs reviewed in Section 2.3,
- evaluation methodologies and frameworks for VANETs outlined in Section 2.4, and
- data sources for content disseminated via VANET messages discussed in Section 2.5.

In general, there is a very high amount of literature in regard to VANETs. A meta-survey providing an overview of the numerous surveys in this domain is given in [272]. Thus, the following discussion is limited to the concepts treated in detail in the later chapters of this thesis, to keep the presentation compact and to avoid overloading it.

2.1 VANET Basics

This section provides an overview of fundamentals in regard to basic VANET mechanisms and requirements. A general overview about VANETs' state of the art is also given in [87]. Targets of the VANET approach and resulting requirements are looked at in Section 2.1.1. The basic communication protocol architectures from European and US approaches for VANET deployment under the outlined requirements are given in Section 2.1.2.

2.1.1 Targets and Requirements

The primary goal of VANET deployment is the support of cooperative driving. This means VANETs are intended to enable safety-critical data exchange, on which future ADAS are based. Thereby, an increase of safety of driving should be achieved. Additionally, non-safety-critical communication within VANETs, for several new kinds of comfort and entertainment purposes

for drivers and passengers, has been proposed [59, 77, 287]. Such communication patterns are typically IP-based, while dedicated VANET protocols have been developed for safety-critical communication. The focus on this work is on safety-critical communication within VANETs. Hence, aspects of non-safety-critical communication are not treated in detail in the following, and the reader is referred to, e.g., [59, 77, 287] for more details about other kinds of communication within VANETs.

To increase safety of driving, collision avoidance is a central goal of cooperative ADAS, i.e., ADAS being (partly) based on data communicated via a VANET. High node mobility and limited communication range of each node pose challenges to such communication. Hence, connection times of nodes are often relatively short. Thus, protocols requiring long connection setup times are inappropriate for VANET communication patterns. In contrast, all communication should be as stateless as possible, i.e., each message should be usable on its own, to minimize delay between first radio contact and actual application data exchange [59, 77, 287].

Furthermore, a high level of availability is required for data exchange in VANETs. Hence, many approaches facilitate a decentralized realization, i.e., there is no dependency on any extra infrastructure to enable communication. However, additional infrastructure in the form of road side units (RSUs) should be supported [59, 77, 287]. In contrast, other kinds of proposals realize V2X communication based on the infrastructure provided by cellular communication networks, e.g., [12, 98, 189, 259]. Both kinds of approaches share common challenges, especially in regard to functionality on and above the network layer as well as security mechanisms. Hence, many of the results presented in this work also apply to V2X communication systems based on cellular communication networks, while the focus of this work is on the decentralized approach without infrastructure support.

Moreover, data exchange should be robust against distortions on the wireless channel caused by high node mobility. Hence, adaptations of wireless physical layer approaches dedicated to VANETs have been developed. The most popular one is probably IEEE 802.11p, a variant of the well known IEEE 802.11 standards family, alongside with its very similar European counterpart ITS-G5 [59, 106, 169].

In general, accurate position and time information is required within VANETs [59, 285]. This need emerges from the requirements of applications as well as protocol functionality like routing algorithms. A detailed look at the security aspects of these basic data sets is provided in Section 2.5.2.

European and US approaches for realization of VANETs based on the requirements outlined in this section are discussed in the following section.

2.1.2 Communication Protocol Architectures

Basic information dissemination in VANETs is done by broadcasting cyclic messages, which are often called beacon messages, or shortly *beacons*. Corresponding messages for ETSI ITS and WAVE are called Cooperative Awareness Message (CAM) and Basic Safety Message (BSM), respectively. Data sets, which are only distributed on demand, are handled differently in ETSI ITS and WAVE. While the dedicated Decentralized Environment Notification Message (DENM) is used in ETSI ITS, WAVE extends the BSM by appending an extra data container [59].

The communication protocol stacks within ETSI ITS and WAVE share many features. However, there is significantly more difference at higher layers, from the network layer upwards, while lower layers and the applied security mechanisms are pretty similar [59, 272].

A general overview of the protocol stack architectures of WAVE and ETSI ITS is given in the following. Details about the chosen security approach are discussed in detail in Section 2.2.

2.1.2.1 WAVE Architecture

The WAVE approach for VANET realization provides a dedicated protocol stack for safety-critical communication [59, 77, 174, 176, 307, 324]. An extension towards a protocol stack supporting IPv6 based communication is available, but is not considered in the following as this topic is not in the focus of this work.

An illustration of the WAVE protocol stack is given in Figure 2.1. It shows that two cross-layer entities exist, which are the WAVE Station Management Entity (WSME) as well as the security entity. Network and transport layer functionality is combined within the so called WAVE Short Message Protocol (WSMP). In comparison to IP-based protocol stacks only a very limited amount of functionality is provided. For example, there is no support for multi-hop communication, i.e., WSMP does not provide routing support for messages.

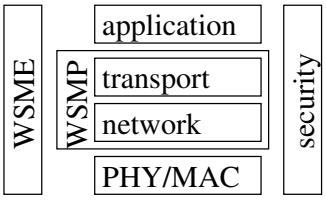


Figure 2.1: Architecture of the WAVE protocol stack for safety-critical communication.

The PHY/MAC layers of WAVE are defined by the 802.11p standard, which is a variant of the well known 802.11 standards family with dedicated optimization for communication in vehicular environments [169]. The interaction of individual layers with the security entity is treated in detail in Section 2.2.4.

2.1.2.2 ETSI ITS Architecture

Within ETSI ITS the protocol architecture illustrated in Figure 2.2 is used for safety-critical communication [59, 101]. As for WAVE, an extension towards IPv6 is available, which reuses PHY/MAC as well as network layer functionality from safety-critical communication [59, 123]. It is intended for non-safety-critical communication. Thus, the reader is referred to [59, 123] for more details, as this kind of VANET communication is not in the focus of this work.

The cross-layer ETSI ITS security system is proposed in [25]. It is described in detail in Section 2.2.4. The second cross-layer entity is the management entity [101]. It realizes, e.g., the cross-layer Decentralized Congestion Control (DCC) management [124].

A summary of ETSI ITS standards and corresponding extensions for their implementation by the C2C-CC is given in [54]. However, many standards are contained in an outdated version.

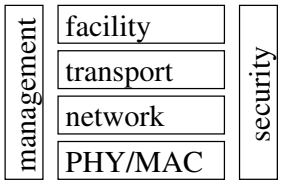


Figure 2.2: Architecture of the ETSI ITS protocol stack for safety-critical communication.

Moreover, a general overview of the ETSI ITS approach is given in [138].

The PHY/MAC layers for ETSI ITS are given by the ITS-G5 standard [106]. In general, ITS-G5 is similar to 802.11p, but some parameters have been adapted for the European approach. For example, the concept of Decentralized Congestion Control (DCC), which is not present in WAVE, leads to extensions of the Medium Access Control (MAC) layer. In doing so, ETSI ITS DCC rules specify a maximum channel access time for the ITS-G5 physical layer. Thereby, a maximum message size at this layer's data input of 650 bytes is enforced [103]. General information about DCC is available in [50, 103].

An overview of the remaining higher level protocol layers is given in the following alongside with references to similar concepts developed in prior work. This covers the network, transport and facility layers.

Network Layer / GeoNetworking Network layer mechanisms within ETSI ITS are often referred to as GeoNetworking. Main mechanisms are standardized in [122]. The term GeoNetworking relates to the feature of position based routing, which is supported by the ETSI ITS network layer. This feature is required to support multi-hop communication in VANETs.

A large number of forwarding mechanisms in VANETs has been suggested [47, 59, 140, 141, 161, 287]. These approaches can be differentiated based on the way the forwarding node is selected. Two major approaches are

- local selection of the forwarder at the sender of a message, and
- decentralized forwarder selection from the set of all possible forwarders.

Sender based forwarder selection is typically done in a way to optimize at least one forwarder selection criterion. Hence, such kind of approaches are often referred to as greedy forwarding schemes. In contrast, decentralized forwarder selection performs the optimization in a distributed way within the network. All nodes receiving the to be forwarded message decide on whether to forward it or not. A popular approach, which is also used in ETSI ITS [122], is contention-based forwarding (CBF) [141]. It uses a combination of timeout and distance based forwarder selection. For more details the reader is referred to [59, 122, 141].

Like in the WAVE system, packet fragmentation is not supported by the current GeoNetworking approach [122, 174]. Approaches for its support have been looked at, but they have not been regarded for standardization (yet) [62, 158]. This leads to a cross-layer data size dependency, as shown in Section 4.3. Such kind of dependencies have not been studied in prior work in the VANET domain. For IP-based communication, the need to consider cross-layer maximum size limitations during protocol design is stated in [14, 66, 319]. The provided analysis in

Section 4.3 shows that one has to massively limit data size requirements of current ETSI ITS facility layer messages and/or the security entity's security envelope. Otherwise, violation of maximum packet size limitations at the network and/or MAC layers occurs. This problem can be addressed by a cross-layer management for inclusion of sporadically disseminated data sets.

Transport Layer Only a single transport layer protocol for safety-critical VANET communication within ETSI ITS has been proposed so far. It is called Basic Transport Protocol (BTP). BTP provides just two port numbers to separate messages of higher level applications, i.e, those on the facility layer. Setting of the sender port number is mandatory. In contrast, it is optional to provide a dedicated receiver port number [122].

BTP shows similarities to well known User Datagram Protocol (UDP) [207], but with reduced protocol overhead. UDP additionally provides the length of the higher level messages as well as a corresponding checksum [207].

Facility Layer The basic information exchange on the facility layer is handled by the so called Cooperative Awareness Message (CAM). Additional on-demand information dissemination is performed using Decentralized Environment Notification Messages (DENMs). Moreover, dedicated messages have been considered for various use cases, e.g., signaling of traffic light phases [186]. However, it has not been decided how such messages should be included into the ETSI ITS approach, e.g., which channel should be used for their dissemination. Each message type is handled by a dedicated so called basic service, as illustrated in Figure 2.3.

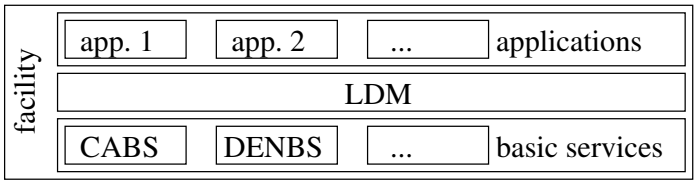


Figure 2.3: Parts of the facility layer in ETSI ITS.

CAMs are handled by the Cooperative Awareness Basic Service (CABS), while DENMs are treated by the Decentralized Environment Notification Basic Service (DENBS). These basic services receive the corresponding messages from the transport layer and are also responsible for sending of messages based on triggering conditions, e.g., cyclic message dissemination [119, 120].

In general the set of all available data structures within facility layer messages is defined in the so called Common Data Dictionary (CDD). In detail definitions of these data sets can be found in corresponding standards [117, 119, 120].

To store received messages for later usage by applications, a common data container called Local Dynamic Map (LDM) has been introduced [118]. Several extensions of the LDM concept have been proposed in order to put more intelligence into it than just pure data storage [160, 195, 285].

2.2 VANET Security Architecture

Security and privacy aspects are core concerns during the development of VANET realizations. General overviews about VANET security are provided within [202, 210, 251]. A more general look at vehicular security is given [327], which also briefly covers VANET security alongside with in-vehicle security. An overview of security aspects in regard to wireless ad-hoc networks is given in [80], which shows that many challenges and corresponding approaches are quite similar for different kinds of such networks, e.g., VANETs. However, a number of individual characteristics exists for the dedicated realizations, which is caused by their specific use cases [80].

The following sections give an overview about various topics in regard to VANET security. At first, basic requirements and used cryptographic mechanisms are looked at. Then, the privacy protection approach taken in ETSI ITS and WAVE is studied. Afterwards, a close look on various implementation aspects of the general design from prior sections is provided in Section 2.2.4. The general focus is on topics studied in detail in later chapters of this thesis.

2.2.1 Requirements

Main requirements for security mechanisms in VANETs are [89, 100, 188, 279]

1. data integrity,
2. data authenticity,
3. access control, and
4. privacy as well as accountability of drivers.

The requirements of integrity and authenticity are typically realized via a PKI scheme and securing messages with digital signatures (see also Section 2.2.2.1). Privacy of drivers is ensured by dedicated privacy protection mechanisms, which have been developed for VANETs, as explained in Section 2.2.3. Strict privacy protection mechanisms in VANETs are especially demanded in Germany [147].

Moreover, total anonymity of nodes (resp. drivers) is not desired in VANET realizations. Instead, a pseudonymization scheme is preferred to allow on demand pseudonym resolution in order to react to misbehavior of VANET participants. Thereby, the requirement of accountability is realized. Closely related to accountability is the requirement of access control, i.e., there is only a well known set of legitimate nodes in a VANET. In current VANET approaches this requirement is realized via the mentioned PKI scheme, which only grants Pseudonym Certificates (PSCs) to well known members of the VANET [89, 176].

An additional requirement affecting security mechanisms is availability. DOS attacks typically try to force a target to become unavailable. Such attacks are a quite popular kind of attack in regard to VANETs, as discussed in more detail in Section 2.3.2. However, availability is a more general requirement, which also includes aspects from various other layers, e.g., robust channel coding to cope with distortions on the wireless channel [295].

Furthermore, some applications require confidentiality of data exchange by content encryption. However, the majority of data dissemination within current ETSI ITS and WAVE approaches use plain text dissemination of data sets. Section 6.6 is dedicated to data encryption within ETSI ITS. It is shown, that the current approach suffers from a major design weakness, which has not been identified in prior work. The current way of handling data encryption within the security entity does not allow to combine data encryption with multi-hop communication. A proposal to overcome this problem is provided.

Finally, there are also non-security related requirements for security functionalities within VANETs [185, 206, 275, 279]. These mainly affect the size of extra data sets, which have to be exchanged to enable security mechanisms to work, e.g., digital signatures and certificates. This is caused by massive bandwidth restriction within VANETs [185,206,275,279]. Moreover, computational requirements of chosen algorithms should be limited, due to the need of limited energy consumption of vehicles and cost sensitivity of the automotive domain [192, 206, 279].

The mentioned requirements can also be found in the Mobile ad-hoc network (MANET) domain [84], and other kinds of wireless ad-hoc networks [80]. However, approaches for Wireless Sensor Networks (WSNs) and MANETs typically not consider a multi-level PKI, i.e., distribution of intermediate CA certificates is not required [145, 260]. Moreover, WSN proposals assume a-priori certificate dissemination, i.e., there is no on-demand delivery of new certificate to nodes within the network [145, 260]. In contrast, these issues have to be considered for VANETs [125, 176].

To fulfill the given requirements the security entity requires knowledge of accurate time and position information. This topic is discussed in more detail in Section 2.5.2.

2.2.2 Cryptographic Mechanisms

Cryptographic mechanisms for usage within ETSI ITS and WAVE have been standardized in [125, 176]. Prior work on general efficiency of cryptographically secured protocols typically addresses point-to-point communication and/or focuses on runtime performance [13,16]. In contrast, broadcast authentication with efficient bandwidth usage and without a lengthy connection setup procedure is required in VANETs for safety critical use cases. Details regarding digital signature schemes and encryption mechanisms utilized in VANETs are given in the following.

2.2.2.1 Digital Signatures

Digital signatures for VANETs should provide a high security level with relatively short signature sizes. Thus, the Elliptic Curve Digital Signature Algorithm (ECDSA) has been selected for usage in current standards. General basics of Elliptic Curve Cryptography (ECC) can be found in [194, 225]. Details about ECDSA are given in [244]. ECDSA realizations require a secure hash function to obtain the value over which the signature is calculated. Current VANET standard use SHA-256 [235] for this purpose. Arguments for selection of ECDSA as the signature algorithm for VANETs can be found in [206, 279].

Other digital signature schemes have been proposed for VANETs, e.g., based on the TESLA protocol [165]. However, these have been found to show drawbacks in comparison to the ECDSA approach [275, 279]. This is similar to research results from the WSN domain [250].

2.2.2.2 Encryption Mechanisms

To provide data encryption ETSI ITS and WAVE use the Elliptic Curve Integrated Encryption Scheme (ECIES) [244] in connection with the Advanced Encryption Standard (AES) cipher [234]. Thereby, data confidentiality is provided by symmetric encryption of the confidential payload, while the corresponding key is protected by asymmetric encryption. Efficiency of the chosen encryption approach has not been looked at in detail, in contrast to the area of WSNs [144]. However, the number of exchanged encrypted messages within safety critical VANET communication can be expected to be small [279].

A major use case for confidential data exchange in VANETs is on-demand requesting and reception of PSCs from a backbone located CA. This mechanism is studied in more detail in Section 6.4.1.

2.2.3 Privacy Protection Mechanisms

Privacy protection in VANETs is commonly realized by pseudonymisation of node identities. To avoid tracking of nodes over long distances used identities are change frequently. General work on protection of privacy of nodes in VANETs is given in [167, 205, 251].

To implement the pseudonym approach in VANETs, a coupling of unique protocol identifiers on all layers of the protocol stack has been introduced. A common "master" pseudonym identifier is determined by the security entity. Typically, the PSC is used to obtain such an identifier by calculating its hash value from a secure hash function, e.g., SHA-256. Other protocol layers derive their individually required protocol identifier from the master pseudonym identifier. This scheme is illustrated in Figure 2.4 with ETSI ITS nomenclature (WAVE is similar). Every time the master pseudonym identifier gets changed, all derived identifiers are changed, too. Individual standards for all required interfaces between protocol entities have been published for ETSI ITS [114–116]. The first two bytes of the GeoNetworking address are set in dependence of static node properties. For details see [122]. The so-called station ID is used by the facility layer for node identification [119].

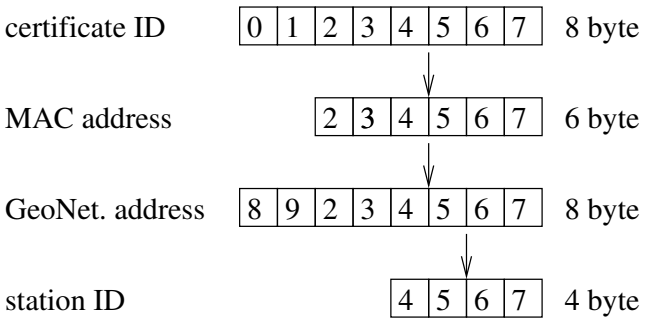


Figure 2.4: Dependency between identifiers from different protocol entities in ETSI ITS.

A pseudonym change strategy determines when and how to change a node's pseudonymous identity. Many different strategies have been developed. These include [92, 251, 294]

- fixed pseudonym change intervals,
- randomized pseudonym change intervals,
- time and location based pseudonym change,
- silent periods in combination with any one of the other strategies,
- MixZones,
- context aware pseudonym change, and
- cooperative pseudonym change.

A MixZone is an extension of the silent period concept, which limits pseudonym change to well defined areas [83, 251]. Moreover, various combinations of the given schemes have been proposed [251]. A common assumption of these mechanisms is that there is no extra information, which enables an attacker to connect the identity before and after a pseudonym change. However, such information is massively present in various data sets on different protocol layers within current VANET standards. This is outlined in detail in Section 5.4.

Negative impact of pseudonym change on reliability of cooperative ADAS has been shown in [205]. This holds especially, for long silent periods and/or large MixZones.

The performance of privacy protection mechanisms can be evaluated by different metrics. Popular metrics are the distance or time an attacker can continuously track a vehicle, and the concept of anonymity sets [251]. An anonymity set holds all nodes which an attacker cannot distinguish. There is more anonymity for a node in case the anonymity set it belongs to has more members [94, 209, 283]. PSC change algorithms have been proposed, which try to maximize privacy of nodes by maximizing the size of anonymity sets [63]. The metric of anonymity sets is used in Section 5.4 to show the negative impact on node privacy caused by data sets being constant and characteristic for an individual node.

Usage of exposed data sets, which are constant and characteristic for an individual node, to perform fingerprinting for node identification has not been studied in-detail for VANETs in prior work. Moreover, there is only a small amount of prior work on that topic in the communication domain in general [91, 313]. Prior work within the Internet domain has focused on tracking users based on web browser characteristics [313]. In contrast, radio fingerprinting, which utilizes the shaping of emitted radio waves, is a well studied subject in the domain of wireless communication [92].

2.2.4 Implementation of Security Mechanisms in ETSI ITS and WAVE

In general, the implementation of security functionality for individual messages is pretty similar within ETSI ITS and WAVE [125, 176]. Both approaches secure messages on the network layer, by embedding their higher level payload and parts of the network layer header fields into a so

called security envelope. A general overview about the implementation of security mechanisms within ETSI ITS is given in [291].

Figure 2.5 shows the relation between network layer meta data and payload to the security envelope. One can see that only a part of the network layer data is secured by being contained in the payload of the security envelope. Another part remains outside the secured data set. Details about the content of the security envelope are given later on (see Section 2.2.4.2).

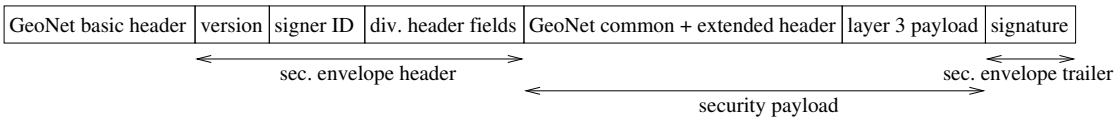


Figure 2.5: Security envelope in relation to network layer header fields within ETSI ITS.

The inclusion of the security entity into the overall protocol stack is illustrated in Figure 2.6 with ETSI ITS nomenclature. The above described securing of messages at the network layer leads to the introduction of a "security layer" between the advanced functionalities of the network layer, e.g., message routing, and basic network layer functionality. In ETSI ITS this basic mechanisms consist of parsing the basic header, while handling all other kinds of network layer headers is subject to the advanced functionality.

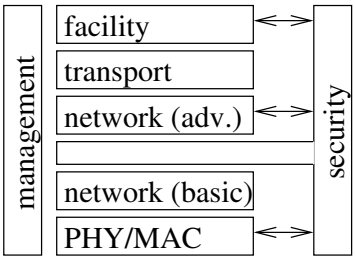


Figure 2.6: Inclusion of the security entity in ETSI ITS from a functional point of view.

Interaction between the MAC layer and the security entity only consists of updating the current MAC address in case the used PSC is changed, i.e., this interface is only used during a pseudonym change [114]. A similar interface exists between the network and facility layers and the security entity, for setting their individual node addresses as well [116]. A more detailed discussion of pseudonym changes is given later on in Section 2.2.4.5.

An extra interface exists between the facility layer and the security entity. By using a so called ITS-Application Identifier (AID) within a PSC, the PSC is limited in validity to the application(s) with present ITS-AIDs. However, the security entity cannot know which kind of facility layer message is present within the payload of the security envelope. Hence, the security layer passes the PSC-ID of a verified message to the network layer (advanced part). This information is passed via the transport layer to the facility layer. There it is passed again to the security entity together with the found message type. Finally, the security entity checks whether the PSC is valid for the found message type and informs the facility layer about the check's result [115, 125].

2.2.4.1 Public Key Infrastructure

ETSI ITS and WAVE security approaches use a Public Key Infrastructure (PKI) [105, 125, 176, 229]. The number of certificate hierarchies is fixed to three within ETSI ITS [105, 125]. In contrast, there is no such limitation within WAVE [176].

The lowest level certificate used in a VANET PKI is commonly called an Pseudonym Certificate (PSC) (or authorization ticket in ETSI ITS). This name emerges from the fact that PSCs are used as temporary identities of nodes, which are changed rapidly to protect privacy of nodes (see also Section 2.2.3). PSCs are issued by an Certificate Authority (CA), which is called an Authorization Authority (AA) within ETSI ITS. Hence, the certificate of such a CA (or AA) is called an Authorization Authority Certificate (AAC). WAVE uses the term Pseudonym Certificate Authority (PCA) for such kind of CAs [154]. Certificates of ordinary CAs are issued by a root CA, which itself generates its own root CA certificate. Hence, a root CA certificate is typically self signed. Additionally, cross signing of root CA certificates may be used [105].

Corresponding PSC, AAC and root CA certificate represent the so called certificate chain used by a node. Current standards assume that all used root CA certificates are known to all nodes. Hence, there is no mechanism to distribute them on-demand in the VANET. In contrast, all other certificates are disseminated on-demand using mechanisms described in Section 2.2.4.3.

The PKI concept is well known from the Internet domain [193, 332]. An important example for its usage is the Transport Layer Security (TLS) protocol [82]. In contrast, MANET research commonly focuses on infrastructure-less concepts without a-priori fixed CAs and corresponding PKI schemes [74, 84, 159, 184, 333]. This also holds for other kinds of wireless ad-hoc networks [78]. In general, MANET requirements are very similar to the ones from VANETs in regard to security aspects, as mentioned in Section 2.2.1. Hence, the results on PKI based security mechanisms obtained for VANETs can be expected to be well portable to MANETs.

2.2.4.2 Security Envelope

The full definition of the security envelope for ETSI ITS and WAVE is available in corresponding standards [125, 176]. Figure 2.5 shows the overall structure of the security envelope with ETSI ITS nomenclature.

The data sets present within the security envelope depend on the security profile chosen by higher level functionality. Within ETSI ITS dedicated profiles for CAMs and DENMs alongside with a generic profile for all remaining messages have been defined [125]. The focus in this work is on the security profile for CAMs, as the majority of data exchange in ETSI ITS happens via this message type.

In general, the security envelope consists of a header holding multiple header fields, the payload and the trailer. The standard allows the presence of multiple trailer fields, but only one type of trailer field has been defined so far. It holds the signature of the message, which is calculated over the header fields as well as the payload [125, 176]. All the individual header fields are discussed in Section 5.4.1.2 alongside with their impact on privacy of nodes.

The signer (and sender) of a message is identified by either including his full PSC (also called authorization ticket in [125]) or just a hash value of the PSC (so called HashedId8), which serves

as a PSC-ID, into the security envelope. Both values can be put into the so called *SignerInfo* field (Signer ID in Figure 2.5). In specific cases the certificate chain is put into this field (see Section 2.2.4.3), which also contains the full PSC.

The inclusion frequency of PSCs into messages is commonly limited to limit the channel load, i.e., there is only sporadic inclusion of the PSC into the security envelope. The contribution of a certificate to the entire length of a message is significant (see e.g., [125, 176]). Thus, replacement of a PSC by its just eight byte long identifier yields massive reduction in message size. However, this replacement strategy causes the security entity be the only non-stateless part of the protocol stack for single-hop communication. Message verification requires knowledge about the full PSC. Hence, PSCs have to be buffered and looked up, when their IDs are received later on. In case the full PSC is unknown, the message is discarded, as it cannot be verified. This process is called cryptographic packet loss [133, 135].

The main difference between ETSI ITS and WAVE security envelope formats is the usage of implicit certificates. These are only used within WAVE and provide much shorter certificate representations, which helps to keep average messages size short. Moreover, cyclic inclusion of the pseudonym certificate in the security envelope happens with 2 Hz frequency in WAVE, while ETSI ITS uses just 1 Hz.

2.2.4.3 Certificate Dissemination among Nodes

As outlined above, all elements of the certificate chain used to secure VANET messages, except of the root certificates, are distributed among nodes in an ad-hoc manner. In the following, we divide the discussion to concepts for PSC distribution and dissemination of higher level CA certificates. PSC distribution has been well studied in prior work. In contrast, distribution of CA certificates in VANETs is a hardly studied topic.

Distribution of Pseudonym Certificates Many different strategies for ad-hoc distribution of PSCs in VANETs have been proposed [125, 133, 135, 176, 185, 191, 275]. The strategy used in ETSI ITS and WAVE standards is illustrated in Figure 2.7 with nomenclature from ETSI ITS.

The standardized approach for PSC distribution can be regarded as the result from prior work given in [185, 191, 275]. Hence, such work is the basis for the further development provided in this work.

Alternative approaches for PSC dissemination by RSUs and not by mobile nodes have been proposed [6]. However, such proposals have not been considered in current VANET standards, as they heavily rely on infrastructure, which is not (yet) available.

Recent work argues against the usage of neighborhood based PSC emission. Instead, a channel load dependent PSC omission strategy is proposed. It follows the approach that a PSC is included as often as possible, while limiting the channel load to a well defined maximum by suppressing dedicated PSC emissions [133, 135]. However, this approach suffers from a number of shortcomings, as outlined in the following. These include,

1. increased capabilities of a single static attacker. An attacker who creates bogus channel load can cause (very) low emission frequencies or even no PSC emission at all. Thus, all message exchange with new neighbors will fail, due to cryptographic packet loss.

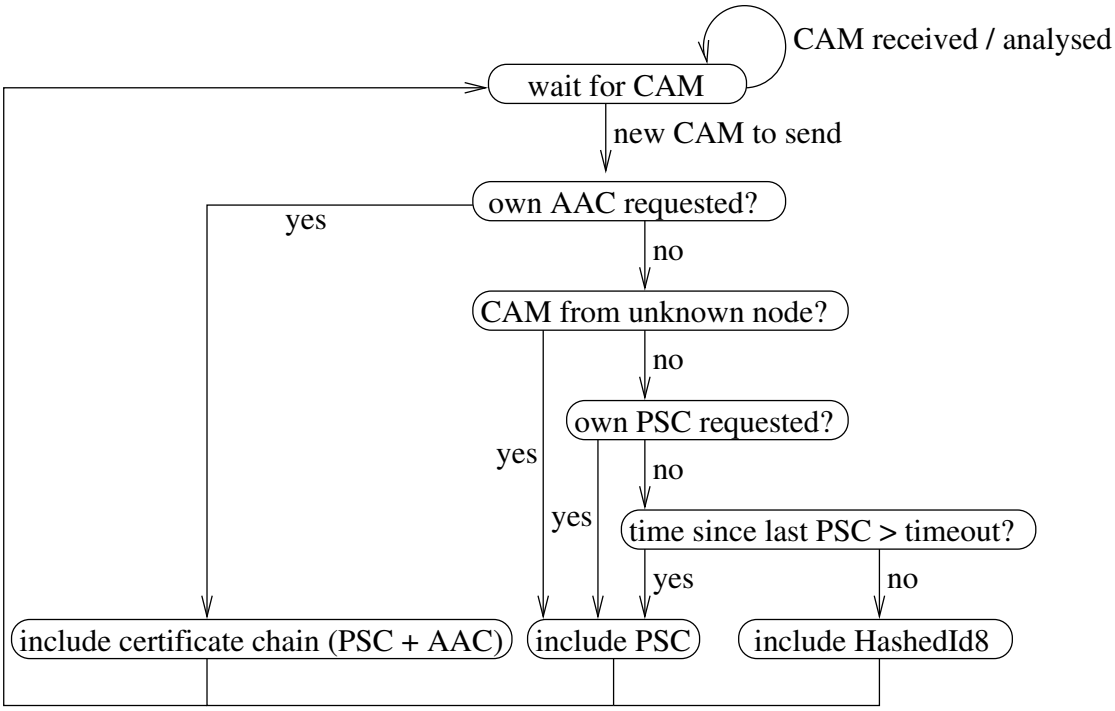


Figure 2.7: Decision process for certificate (chain) inclusion for a CAM or BSM.

2. The approach assumes well PSC distribution in the close environment of a vehicle, but

   - this is not the case in systems using rapid pseudonym change,
   - in urban scenarios with much shadowing due to buildings and other obstacles together with complex road architectures like many junctions, new neighbors can appear in very close vicinity of node, e.g., in the area of some meters.

   in these scenarios rapid reaction to the appearance of new neighbors is required, as they are possible collision partners within a short time frame.

3. The omission approach assumes that the acceptable channel load caused by beacon messages is constant. However, this does not hold in a VANET distributing messages of more than only a single type on the same channel, which is used for beacon distribution. E.g., ETSI ITS distributes CAMs and DENMs on the same wireless channel. DENMs are sent on demand, i.e., it is not clear when and how many nodes transmit such a kind of message. Hence, the bandwidth requirement for the distribution of this message is unclear in advance. Moreover, these messages are used for highly time critical information distribution and disseminated using multi-hop delivery within a relevance area, which can greatly exceed the communication range of a single node. The used CBF strategy greatly suffers from increased channel load. Therefore, one can assume a negative impact of DENM dissemination performance from the channel load increase caused by the certificate omission strategy.

4. The omission approach influences the DCC strategies of ETSI ITS. The accepted channel load in [133, 135] is higher than DCC congestion limits [103]. This affects the behavior of an ETSI ITS stack in two ways. These are

   - a limitation of the CAM generation interval. In doing so, the amount of generated CAMs per time interval can get limited. This affects especially highly dynamic traffic scenarios, because they cause high CAM generation rates (without DCC inducted limitations) [119]. Higher CAM sending intervals reduce the data update rate of applications at receivers, which may lead to worse performance of such applications.
   - Multi-hop delivery of messages, e.g., DENMs, is disabled in all DCC states except of state RELAXED, i.e., a channel busy ratio (CHBR) lower than 15% [103].

   This shows that a negative impact on use cases can be expected, when the PSC omission approach gets combined with ETSI ITSs' DCC mechanisms.

Overall, the certificate omission approach is similar to DCC's CAM omission approach. However, with certificate omission, the security entity is used to fix the problem of too high channel load, which is caused by high higher level message generation frequency. Very high node density is used for system evaluation in [133, 135], together with 10 Hz BSM sending frequency. However, such high traffic densities are typically partly caused by nodes with low mobility, e.g., within a traffic jam. Therefore, the adaptive CAM generation frequency of ETSI ITS already counters the found channel overloading problem from [133,135], by using longer message transmission intervals in comparison to WAVE. Additionally, low mobility of nodes typically does not require high beacon exchange rates. Thus, it seems more appropriate to counter the channel load problems identified in [133, 135] by applying the CAM generation rules (or a similar mechanism) also for BSMs, instead of using a non-neighborhood aware certificate distribution strategy.

In contrast to the PSC omitting approach, an extension to standardized PSC distribution is proposed in Section 4.2. In doing so, we use some of the concepts from [133, 135] to model a node's environment leading to context aware PSC distribution. The omission scheme cannot know were the authenticated and unauthenticated nodes are in its environment by just looking at the channel load. In contrast, we model such distribution by using techniques proposed in [133, 135] to evaluate system performance.

Within the WSN domain the need to adjust security mechanisms for short average message sizes has been looked at as well. In contrast to VANETs, the main motivation for WSNs is energy consumption, as transmission of longer messages significantly adds to the overall energy consumption of sensors. However, efficient certificate distribution has not been looked at, as the focus of research has been on efficient signature and encryption mechanisms [144, 145, 250, 260]. WSNs could profit from the sparse certificate distribution approaches from VANETs, especially as from the ones using new neighbor detection mechanisms, as the mobility of sensors is typically much lower in comparison to the one of vehicles [80].

In case of WAVE, the PSC inclusion mechanism is the same as in ETSI ITS. However, certificate chain distribution is more complex in WAVE, due to an arbitrary number of certificate hierarchies.

Certificate Chain Distribution The distribution of certificate chains in VANETs is a topic hardly covered in prior work. Instead, the focus has been on the development of alternative PKI schemes with the CA being established within the VANET itself on-demand [221, 282]. However, such schemes cannot provide the level of security from fixed and backbone based CAs [221, 282]. [229] assumes that all CAs' certificates are distributed by infrastructure nodes, i.e., RSUs, to all mobile nodes. However, the lack of availability of RSUs clearly limits the possibility to realize this approach.

Work within similar domains, like MANETs, also focuses on fully decentralized PKI approaches, i.e., CAs are established within the network itself [74, 159, 184, 333]. In contrast, the approach in [254] requires a connection to the backbone every time a new connection between new neighbors in the network is established. However, this would violate the requirement of a VANET being able to operate without a permanent connection to a backbone service.

In the internet domain certificate chain distribution is a regular task within the TLS protocol's handshake [82]. However, only the most straight forward solution of always sending the full certificate chain to every communication partner every time a new connection has been established has been considered so far [82]. This is clearly a highly bandwidth consuming approach, which is infeasible for VANET with many communication partners and often short connection times.

The latest revision of the ETSI ITS security envelope standard (from [109] to [125]) introduced distribution of CA certificates among nodes. However, we find the straight forward approach taken in the standard shows significant drawbacks, especially in regard to experienced channel load (see Section 4.2.2 for details). Hence, a detailed introduction of the certificate chain distribution mechanism from ETSI ITS is given in the following.

Certificate chain distribution in ETSI ITS happens by on-demand inclusion of the full certificate chain into the security envelope of a CAM. A typical message sequence leading to certificate chain emission is shown in Figure 2.8 with only two affected nodes. In the shown example, node A is not aware of AAC_B, i.e., the AAC used by node B. In contrast, node B knows about AAC_A in this example. The message exchange is started when node A receives the first message from node B. Afterwards, node A requests the unknown PSC_B and emits its own PSC_A, due to triggered new neighbor detection. However, node A cannot verify PSC_B, because it lacks knowledge about AAC_B. Thus, a request for AAC_B is sent within the next CAM. This leads to transmission of the certificate chain of node B.

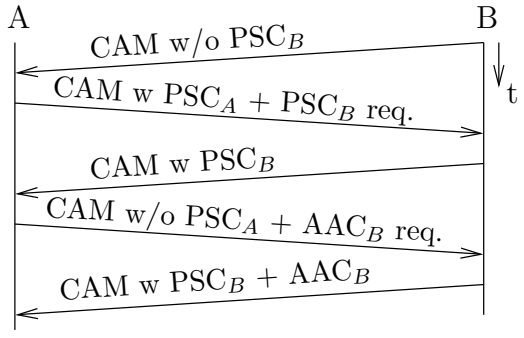


Figure 2.8: Message sequence leading to an AAC request and delivery.

In general, a multitude of nodes can receive a node's request for an AAC, as illustrated in Figure 2.9. This is due to the broadcast nature of CAM dissemination. Moreover, the request is not targeted in any way towards a dedicated node. Thus, all nodes receiving the request and using the requested AAC themselves will answer it with transmitting their certificate chain. Hence, assuming a common AAC used by nodes B, C and D in the example depicted in Figure 2.9, nodes C and D additionally send their certificate chains. These transmissions have to be regarded as pure overhead, as the AAC has already been delivered to node A by a prior message from node B.

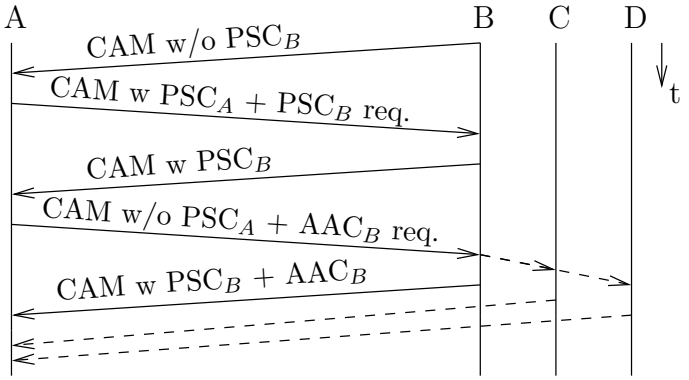


Figure 2.9: Message sequence leading to multiple deliveries of the same AAC.

It can be assumed that many nodes share a common AAC. Thus, the number of answers to an AAC request can be high. However, every delivery of the AAC after the first one is superfluous and creates channel load without any further benefit, as no new information is provided to the recipient.

One should note, that the meeting of two nodes, which share no knowledge about each others certificate chain, except of the root certificates, is a valid use case. Distribution of certificate chains is studied in more detail in later chapters of this work.

2.2.4.4 Certificate Refill inside Nodes

The limited validity time of PSCs leads to the requirement to obtain fresh PSCs from time to time [85, 251, 291]. The refill (or update) frequency depends on the lifetime of PSCs as well as on the fact whether pre-caching of PSCs inside nodes is used. Buffers holding many PSCs inside nodes have been suggested to avoid frequent refills [216, 251]. However, the attack on the time base of VANET nodes provided in Section 5.3.1 shows that such pre-caching of PSCs with future validity times massively endangers security inside a VANET. More details are given in Section 5.3.1. A suggestion of a PSC refill protocol inside VANETs is given in [291]. However, the need of privacy of refill requesters is not fulfilled by that proposal.

The general outline of a PSC request from a node to its affiliated CA and the delivery of the signed PSC ready for usage is described in [105] for ETSI ITS. The standardized mechanism can be regarded as an extension of the proposal from [291], which also provides privacy to

requesters by encryption of their requests. The generation of a refill request message according to [105] is illustrated in Figure 2.10. A similar mechanism for WAVE is to be found in [176].

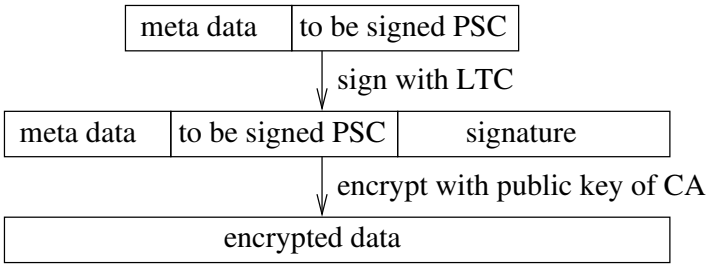


Figure 2.10: Assembling of a PSC request message from a node to its CA.

To protect privacy of the requester, the payload of the request is encrypted using the public key of the CA. The message is first signed with the Long Term Certificate (LTC) of the requesting node, then encryption is performed. This is required as the meta data within the request contains a unique long term identifier of the node, e.g., the LTC itself, within the meta data block shown in Figure 2.10.

The CA answers the request by sending back a message, which contains the signed PSC. It gets encrypted with the public key of the requesting node to protect its privacy.

Prior work has focused on the distribution problem of RSUs with backbone connections to support the pseudonym update process [208, 293, 316, 317]. In contrast, the communication protocol for the PSC request itself has not been evaluated. Such an evaluation is provided within this work. It shows that each request causes high computational load at the CA. Thus, such requests are prone to be used for DOS attacks on CAs. Hence, we propose and evaluate an alternative mechanism to massively limit the computational load, which can be caused by an attacker. For more details the reader is referred to Section 6.4.

Requests for fresh certificates are commonly known as Certificate Signing Requests (CSRs) in the Internet domain [239, 332]. Thereby, privacy of the requester is somehow protected by encryption of the CSR with the public key of the CA, while the signed certificate is delivered encrypted with the public key contained in the certificate itself. Hence, only the requester can decrypt the response from the CA [193, 332]. However, there is no well established mechanism to communicate between requesters and CAs in a privacy preserving way.

2.2.4.5 Pseudonym Change

To protect the privacy of VANET nodes, e.g., vehicles and their drivers, their used pseudonyms have to be changed from time to time. A pseudonym change includes to change all unique identifiers used by all protocol layers, e.g., the station ID used by ETSI ITS facility layer entities [251, 279], as mentioned in Section 2.2.3.

Only changing the pseudonym, by switching the PSC, each time a vehicle is switched on and off is suggested in [54] for ETSI ITS. In contrast, frequent timeout based switching (e.g., every five minutes) during live operation of a mobile node is proposed in [154] for WAVE. Much work has been done to identify ways to perform pseudonym changes avoiding that an attacker

can continuously track the movement of a vehicle [24, 58, 96, 146, 251, 273, 291, 303, 325], as outlined in Section 2.2.3. However, neither ETSI ITS nor WAVE standardize a pseudonym change procedure.

2.2.4.6 Verify-on-Demand

Verify-on-Demand (VoD) is one of the schemes proposed for limiting required computational resources for message verification by selecting only a subset of them for whom cryptographic verification is performed. For all other messages, signatures are not verified. VoD proposes to let applications decide which messages should be verified. In this process, the only considered criteria is whether an application performs a safety relevant operation, e.g., a warning is displayed to the driver, based on a particular received message [199, 320].

Initially VoD was proposed for WAVE. An approach for integrating VoD into ETSI ITS by storing digital signatures in the LDM is given in [195]. However, integration of VoD into ETSI ITS is subject to a longer running discussion, which can be traced in some parts in [127, 128].

Early works on computational performance requirements for ECDSA state that a verify-all scheme cannot be implemented without support from dedicated crypto-processors [199, 279]. Thus, much effort has been put into the development of such dedicated chips [192].

Recently, a number of serious advances for more efficient implementations of ECDSA in software has been published [5]. Some parts of them are already available in the popular OpenSSL library [240]. Corresponding benchmarks for modern processors with moderate speeds show that the number of verifications that can be performed per second greatly exceeds the number of messages, which can be received per second over a highly bandwidth restricted 802.11p or ITS-G5 channel. Hence, systems without possibly expensive dedicated crypto-processors can implement a verify-all scheme. This significantly lowers the necessity of applying VoD.

An overview of security related problems introduced by a VoD approach based on current ETSI ITS standards is provided in Section 5.2. Together with the recent progress of more efficient software implementations of ECDSA, such attacks call the feasibility of VoD into question. Thus, usage of verify-all schemes is recommended for VANETs.

2.2.4.7 Hash Chains for Node Authentication

Hash chains are a popular cryptographic primitive. They have been originally proposed in [203]. The core aim of a hash chain is to provide a sequence of numbers, whose future elements cannot be predicted by an advisory monitoring arbitrary system outputs. For example, such outputs are transmitted from their creator to some communication partner in plain text to be used as an One Time Password (OTP). Many other use cases have been developed as well [21]. In some designs, it is trivial to check whether an element is the correct successor of the last provided element from the chain. This holds for all the proposals from [21, 79, 81, 203, 330]. These schemes can be seen as asymmetric approaches, as only the sender of such a sequence needs to know a private secret, but every receiver can check correctness of the output sequence.

More recent designs (HMAC-based One-time Password Algorithm (HOTP), Time-based One-time Password Algorithm (TOTP)) do not provide this property. Instead knowledge of the

pre-shared secret key is required for verification of the sequence. Thus, these approaches can be called symmetric ones. If used for an entity's authentication, hash chains often fulfill the same purpose as so called hopping codes or rolling codes [15, 97]. Such designs, typically rely on symmetric keys known to sender and receiver.

In the original design from [203], an initial secret key s_0 is iteratively provided to a cryptographically secure hash function j times, as shown in Figure 2.11. This yields exactly $j + 1$ securely usable chained values. These are used in the inverse order of their generation.

$$s^0 \longrightarrow h(s^0) = s^1 \longrightarrow h(s^1) = s^2 \longrightarrow \ldots \longrightarrow h(s^j) = s^{j+1}$$

$$\xrightarrow[\text{generate}]{} \qquad \xleftarrow[\text{use}]{}$$

Figure 2.11: Initial design proposal of a hash chain [203].

While the original proposal yielded a-priori fixed length chains, extensions for infinite length chains were proposed in [21, 79, 81, 330]. In these proposals, typically the hash function from the initial design is replaced with some asymmetric signing algorithm, at the cost that receivers need to know the corresponding public key to perform verification of the chain. Moreover, recent approaches towards secure one-time passwords, like HOTP [230] or TOTP [231], can be regarded as infinite length hash chain designs.

Authentication via a hash-chain based approach is typically much faster than by the usage of asymmetric cryptography based digital signatures [28, 279]. Hence, their inclusion into the PSC refill procedure of VANETs is proposed in this work.

2.2.4.8 Platform Independent Data Representation

Many different kinds of platform independent data representation schemes have been proposed for usage in communication systems. They are used for data exchange between different entities within a network to ensure a common understanding of the contained data sets, which is independent of the dedicated implementation used within individual nodes. For the case of VANETs, the focus on prior work has been on dedicated binary encoding schemes as well as on different variants of Abstract Syntax Notation 1 (ASN.1) [109, 110, 119, 120, 122, 174–176]. However, no in-detail comparison of the performance of different data representation schemes in regard to VANET data sets has been published.

Nevertheless, there are several publications providing evaluations and comparisons of platform independent data representation schemes. Regarded schemes include various variants of both Extensible Markup Language (XML) and ASN.1 as well as JavaScript Object Notation (JSON), Google Protocol Buffers (protobuf) and EXI [90, 95, 149, 150, 311].

In [242] an evaluation of the performance of different binary encoded XML variants, among them ASN.1 variants, is given by running tests on ordinary PC machines. In [232] a comparison of the performance of XML against ASN.1 Basic Encoding Rules (BER) on digitally signed data is provided. The conclusion is that for applications where high performance is required, ASN.1 BER may be a better choice.

In [143] the performance of XML, JSON and protobuf gets compared in terms of data size and coding speed. The authors conclude that protobuf requires less bytes for content representation in comparison with XML or JSON. Moreover, the option to compress XML and JSON encoded data sets using gzip [142] is studied. Both compressed text formats perform better than protobuf in terms of data size. In regard to runtime, protobuf performs better than both plain text schemes [143].

In [148] a similar study to the one in [143] is given. The evaluation is expanded to include energy consumption, which is especially relevant for the considered smartphone use case. Moreover, it is shown that gzip-compressed protobuf, a variant not considered in [143], performs better in terms of encoded data size in comparison with compressed XML, but worse than compressed JSON. In respect to encoding time, protobuf performs better for the considered data set. For the parsing process on the receiver side, i.e., decoding, JSON performs slightly better than the other two schemes [148].

A performance comparison between gzip-XML as well as ASN.1 Packed Encoding Rules (PER) against EXI is provided in [51], where it is shown that EXI greatly outperforms both other schemes for the used test data set. [248] highlights the advantages of schema-enabled EXI in the domain of multimedia applications for embedded systems over the use of plain XML in respect of encoding and decoding performance as well as compression. They especially focus on the encoding of Scalable Vector Graphics (SVG) vector graphics, and introduce an approach for a more efficient data type representation in combination with EXI in this domain.

An alternative suggestion to binary encoding of the security envelope from [109] using ASN.1 encoding has been proposed in an ETSI ITS draft [110]. However, there are no performance studies available providing insights on which alternative should be selected. Moreover, EXI encoding has not been considered for data representation in ETSI ITS so far. Moreover, to the best of the knowledge of the author, there are no previous studies focusing on a quantitative comparison of performance measurements between ASN.1 Unaligned Packed Encoding Rules (UPER), protobuf and EXI. Thus, such kind of evaluation is provided in this work. Thereby, an insight into the overhead caused by platform independent data representation is provided, and a recommendation for the to be used data representation scheme within VANET security functionality is provided.

2.3 Attacks on VANETs

Many different kinds of attacks on VANETs have been proposed [188, 218, 292]. Obtained attacks are based on various attacker models. Hence, an overview about common attacker models is given in Section 2.3.1. Afterwards, different kinds of attacks are discussed in detail. A general overview about attacks on VANETs is given in [222]. Such attacks show many similarities to attacks on other kinds of wireless ad-hoc networks [80].

2.3.1 Attacker Models

A common approach is to differentiate attackers based on the communication area covered by the attacker. Typically, two different kinds of attackers are considered, which are [251, 312]

1. local attackers, which just influence a well defined area around malicious nodes and can be further differentiated into

   (a) static attackers, and

   (b) mobile attackers.

2. global attackers, which have full access to the whole VANET.

Moreover, each of the mentioned attacker types can be either passive, i.e., only receiving messages, or active, i.e., sending and receiving messages. Additionally, insider and outsider attacks can be discriminated. An attacker is considered an insider in case he controls a valid node being part of a VANET. In contrast, an outsider attacker has no physical access to a valid node [251, 312].

2.3.2 Denial of Service Style Attacks

A common requirement of DOS attacks is the presence of an active attacker. The simplest attack on data dissemination in a VANET is jamming of the used wireless channel(s). A detection mechanism for DOS attacks on VANET beacon distribution by jamming is proposed in [214]. The area affected by the jamming attack is mainly limited by the transmission power, which the attacker is able to use. A jamming attack clearly violates regulations in regard to channel usage limitations and the attacker's node massively violates standards for VANET communication. In contrast, this work proposes DOS attacks misusing valid protocol functionality, which are much harder to be detected, as they facilitate only valid VANET messages. Moreover, it is shown that an attacker can target areas, which greatly exceed his own communication range by causing invalid behavior of valid nodes. Details are given in Chapter 5. A general overview of DOS attacks on VANETs is given in [157].

A so called PSC depletion attack is proposed in [251]. To perform it, an attacker sends messages with a duplicated network layer source address to the target, i.e., the attacker uses the same address as the target, to cause a pseudonym change at the target. In case PSCs are only used once, the attacker is assumed to be able to cause the change such often that the target has no more PSC available. However, following the security concept of both ETSI ITS and WAVE, this attack is almost impossible. The check for a duplicated address is performed after the corresponding message got verified by the security entity. Thus, the attack can only be carried out by an insider attacker with access to valid ITS credentials. Moreover, the attacker needs a PSC whose ID (i.e., its shortened hash value) leads to the network layer address (see also Section 2.2.3), which the attacker wants to duplicate. Otherwise, the PSC used by the attacker does not match the network layer address used in the attacker's messages. Hence, a simple consistency check at the receiver would identify the attack and lead to discarding of the attacker's message. Obtaining the required PSC(s) can be assumed to be very hard for any attacker. The attack not even works in case VoD (see also Section 2.2.4.6) is used, as a duplicated address should clearly trigger verification of the message, which caused the detection.

2.3.3 De-pseudonymisation or Tracking Attacks

De-pseudonymisation and node tracking attacks compromise privacy of nodes. Much work has been done in regard to such attacks on VANETs (see also Section 2.2.3).

De-pseudonymisation of nodes re-using PSCs from a PSC pool with low update intervals via long term monitoring by a static passive outsider attacker is shown in [45]. Time triggered PSC changes are monitored and tracking is used to obtain a mapping of different PSC to their common user. A backbone database is created, which allows to learn many or even all PSCs from the pool used by a node. Thereby, pseudonymisation of the node can be circumvented. An extra location dependency of PSC changes can limit the impact of the attack, but more powerful attackers can easily overcome this countermeasure. Hence, re-usage of PSCs is discouraged [45].

Node tracking via characteristic data sets on different protocol layers is discussed in Section 5.4. Moreover, suggestions for improvements of current standards are given to overcome the found weaknesses of the current approaches. Availability of prior work is very limited in regard to this aspect. Within the C2C-CC a still unpublished draft for a privacy memo has been started, which briefly gives some aspects being similar to the analysis provided within this work. However, no in detail evaluation is provided there [274].

2.3.4 Attacks on VANET Applications

Attacks on VANET applications, e.g., ADASs, are typically either based on banned reception of messages (see DOS attacks in Section 2.3.2) or on injection of extra malicious messages. Countermeasures to successful message injection attacks are provided by the digital signature and PKI scheme outlined above. However, usage of VoD enables such kind of attacks, as shown in Section 5.2.

The GNSS spoofing based attack introduced in Section 5.3 allows to perform message injection attacks, which are based on replay attacks. Moreover, we show that advanced attackers can even perform such attacks in the form of a Sybil attack.

2.4 VANET Performance Evaluation

VANET performance evaluation heavily depends on conducting simulations [288]. Real world experiments have been performed, but achieved density of vehicles equipped with VANET technology is limited. Moreover, it is hard to reproduce scenarios with many involved vehicles in practice [285]. Hence, most studies on VANET performance rely on simulation results.

A common feature of discrete event simulators, like the ones used for VANETs, is that they do not take computational requirements encountered at individual nodes into regard to evaluate system performance. This means that the obtained results represent a system in which processing of data at the individual nodes takes zero time. This is caused by the behavior that a discrete time stamp increase only happens after all processing started at the prior time stamp has ended [265, 309]. However, in reality time clearly proceeds, while computational operations are ongoing. Thus, one has to make sure that used algorithms are fast enough for practical usage in a separate evaluation step.

framework	focus	missing standardized features	traffic flow simulator	network simulator
Veins [288]	WAVE	security, ASN.1, time/position coupling	SUMO	OMNET++
Artery [262]	ETSI ITS	time/position coupling no ITS-G5 (802.11p used instead), DCC	SUMO	OMNET++
iTETRIS [155, 215]	ETSI ITS	security, ASN.1, time/position coupling	SUMO	ns-3
VANETSim [302]	security	no standard compliance	own	own
VNS [137]	traffic flow	no standard compliance	own	ns-3, OMNET++
VSimRTI [277]	ETSI ITS	?	SUMO, VISSIM	ns-3, OMNET++
ezCar2X [266]	ETSI ITS	-	SUMO	ns-3
EstiNet [318]	SDN	security, all above network layer	own	own

Table 2.1: Comparison of features of different simulation frameworks for VANETs

Available simulation environments for VANET evaluation are looked at in more detail in Section 2.4.1. Moreover, methods for measuring the computational performance of algorithms are presented in Section 2.4.2.

2.4.1 VANET Simulation Environments

Several simulation environments for VANETs have been developed [137, 215, 266, 277, 288, 302, 318][1]. These vary greatly in regard to standard conformance and the set of supported features. Typically, a combination of different tools for simulation of traffic flow, communication network and protocol behavior is used. This follows the tool coupling approach proposed for VANET simulation in [278]. Popular network simulators are ns-3 and OMNET++ [263, 309]. Microscopic traffic flow simulation is often provided by Simulation of Urban MObility (SUMO) [18]. The VSimRTI framework can also utilize VISSIM [136] for traffic flow simulation. An overview of the features provided by the different frameworks is given in Table 2.1. A similar discussion is provided in the author's prior work given in [43][2].

All of the frameworks from Table 2.1, except of the last three ones (VSimRTI, ezCar2X, EstiNet), are freely available in open source form. Work on further frameworks TraNS, GrooveNet, NCTUns (predecessor of commercial EstiNet) and MobiREAL, all looked at in [88, 220], has

[1]The author's contribution to [266] mainly relates to the design and implementation of security functionality, platform independent data encoding schemes and integration into ns-3. The remaining contributions are from the coauthors.

[2]The coauthor's contribution to [43] mainly relates to the implementation of functionality enabling the conducted tests of the ezCar2X framework. The main contribution is from the author of this work.

been discontinued. Thus, they are not taken into regard in this work. Features of VSimRTI are gathered from the publicly available information about this tool set, as studying the source code is not possible [59, 76, 277].

The Artery framework's feature support is close to the one of ezCar2X. However, due to the usage of 802.11p on the physical and MAC layers no ETSI ITS conforming DCC can be realized. DCC requires channel usage information from the MAC layer [103], which is not available in the Artery approach. Moreover, this also means that the strict MAC layer message size limit from ETSI ITS is not enforced by Artery.

The simulation framework utilized in this work is based on the ezCar2X framework [266, 267][3]. An in detail description and a comparison to other simulation approaches is provided in Section 3.3.

2.4.2 Computational Performance Measurement

Computational performance can be benchmarked with different performance metrics [153, 219]. Important metrics used in this work are runtime and memory footprint (stack and heap). A methodology for accurate time measurement avoiding unintended influence of modern processors out-of-order execution is provided in [246]. Following this methodology, a so called serialization instruction is additionally inserted before and after the code fragment whose runtime is to be measured. Available serialization instructions are processor dependent, e.g., for all Intel processors the *cpuid* instruction can be used [246].

All runtime measurement results provided in this work have been collected using the outlined strategy from [246]. More details about the applied computational performance measurement methodology are given in Section 3.4.

2.4.3 Traffic Scenarios

Many different traffic scenarios have been suggested for usage in the evaluation process of VANETs. A traffic scenarios consists of the road topology and the traffic flow built up by mobile nodes traveling on the road topology. Optionally, RSUs may be present in the scenario, too. Popular road topologies include

- highway [52, 56, 93, 106, 190],
- highway crossing [93],
- rural road [204],
- urban grid [56, 93, 288, 304], and
- urban roundabout [93].

The highway crossing scenario provides high traffic density, but mutual influence between the individual highways is identified as low in [52]. Hence, we do not consider this setup as a separate evaluation scenario. The detailed parameters of scenarios used throughout this work are given in Section 3.2.

[3] For remarks on [266] see footnote 1.

2.5 Security of Data Sources for Security Mechanisms

The security mechanisms looked at in Section 2.2 rely on various data sets, whose contents do not emerge from the security entity itself. To avoid that the security approach for VANET message dissemination gets circumvented, the data sets provided to the security entity inside the protocol stack have to be secured as well. One can separate the discussion into data getting collected only within the node itself, e.g., from local acceleration sensors, and data sets obtained by the help of communication with external entities, which are not part of the VANET. An example, for such an external entity is a GNSS, whose broadcast data is received by nodes to obtain time and position information. Data sets obtained from outside the node can be targeted by external attackers, while those from the within the node itself can only be manipulated by internal attackers. An illustration of typical data sources in a VANET is given in Figure 2.12. Individual parts of this figure are discussed in more detail in the following.

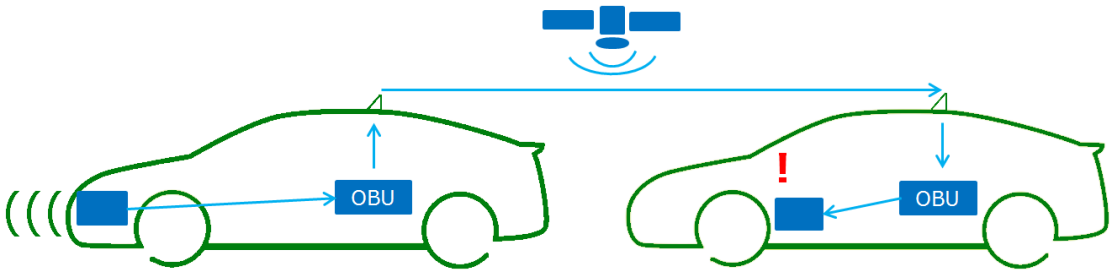


Figure 2.12: Data sources used in a VANET.

Typical data sources from within a vehicle include both vehicle status sensors, e.g., acceleration sensors, and sensors used to monitor a vehicle's surrounding, like radar or lidar sensors. These sensors are typically connected to dedicated control units, which process the sensors' measurement results and provide other control units, like the OBU, with interpreted results over wired connections. For example, a radar sensor can be used to detect a blocked road or a broken down vehicle, and based on that finding the OBU may issue a DENM to warn other vehicles about the detected road hazard [120]. An important data source for an OBU from outside the vehicle itself is GNSS. It is typically used in VANETs to provide a high precision reference time source alongside with location information to vehicles [122, 139, 163, 325]. Furthermore, wireless connections to other nodes in the VANET, e.g., RSUs and OBUs of other vehicles, are key data sources for a vehicle's OBU.

Moreover, an OBU serves as a data source within a vehicle. In doing so, it provides information obtained, e.g., via CAMs and DENMs, to other control units. For example, this can be used to display information about road conditions received from another vehicle to the driver using an human machine interface (HMI).

Security mechanism for data sets emerging from within the node itself are looked at in Section 2.5.1. GNSS input for VANET functionality is discussed in Section 2.5.2.

2.5.1 In-vehicle Security

A popular way for implementing C2X hardware in nodes is to have two (logically) separated devices called Application Unit (AU) and Communication and Control Unit (CCU). The combination of both devices is often referred to as an on-board unit (OBU) [121]. The separation may affect the protocol stack implementation, by realization of individual layers on different devices. A separation above the network layer is suggested in [290]. Thereby, AU holds the higher layers, while the CCU contains realization of the lower layers. Hence, it is required to realize secure communication between AU and CCU. Otherwise, an attacker could misuse the CCU to have its own messages secured, by replacing the AU or injecting malicious messages into the connection of both devices.

Moreover, modern vehicles use many in-vehicle processing units for highly specialized tasks, e.g., a radar sensor as shown in Figure 2.12. These individual Electronic Control Units (ECUs) communicate with each other over one of the several automotive specific bus systems, e.g., Controller Area Network (CAN) or FlexRay [178, 179]. However, the lack of dedicated security mechanisms for such systems leads to the identification of several security issues and corresponding attacks [7, 197, 224, 233]. Hence, proposals for security enhanced versions of the currently used bus systems have been developed [151, 280, 281, 308, 328].

However, prior approaches for security enhancements either extended bandwidth requirements on bus systems or required to move to completely new communication protocols. As an alternative, an approach for efficient securing of typical automotive bus systems without a need to change application layer messages is given in [26, 27]. The given approach works in an (almost) protocol independent way, by reusing the already present Cyclic Redundancy Check (CRC) data fields to store secure signatures. For more details the reader is referred to [26, 27], as this work concentrates on the security of communication outside nodes.

The discussed security approaches can only offer protection about internal attackers targeting the on-board communication systems. Guarding a VANET against data manipulation by attackers targeting local sensors affords additional mechanisms, like discussed in [23].

2.5.2 GNSS Input for VANETs

Basic properties of the very popular GNSS realization Global Positioning System (GPS) are described in [20, 299]. VANETs typically use GNSS input for time synchronization between nodes and to obtain global position information within each node [67, 68, 122, 139, 163, 325]. Unfortunately, the currently deployed GPS does not provide a secured signal for civil purposes. Thus, attacks from simple jamming [180] to advanced spoofing attacks have been developed [168,255,270,299]. A GNSS spoofing based attack on VANET security mechanisms is proposed in this work. In doing so, we reuse existing GPS spoofing capabilities, and build an attack on a VANET's OBUs on top of a successful spoofing attack. The focus of our attack is to manipulate the time synchronization of the attacked OBUs.

Prior work on an attack on a distributed system using GPS time spoofing is provided in [335, 336]. However, the studied SmartGrid system features a static and well known node distribution. This is not the case in VANETs, in which nodes are expected to be highly mobile. Thus, the approach for a countermeasure to GPS spoofing developed in [336] is hardly portable

to VANETs. GPS spoofing is used to attack the time synchronization inside mobile phone networks in [321, 322]. General purpose spoofing detection methods are proposed to overcome the identified threats. An attack on time synchronization of nodes within a static ad-hoc network using wireless frequency hopping communication is described in [334]. The proposed spoofing detection mechanism uses the characteristics of the used frequency hopping scheme. Current VANET approaches do not use frequency hopping. Thus, the spoofing detection mechanism from [334] cannot be used for them. [327] mentions the need to secure time and position information within a VANET node, but no details about how this should be realized are given. [54] specifies the requirement to obtain time and position of a VANET's node in a secure manner.

GNSS spoofing for attacking a VANET is mentioned in [202, 210]. However, no detailed analysis of the impact on VANET security is given. Availability of powerful GPS spoofing methods is demonstrated in [315]. Several general purpose countermeasures to GNSS spoofing have been developed, which allow the receiver to detect the spoofing. These include multi-antenna systems [228] or micro-movement of receiver antennas [256]. A general overview about GNSS spoofing including countermeasures is provided in [181]. However, anti-spoofing techniques are still hardly applied in practice, as they cause high effort, and thus also high costs in implementations [166].

A VANET specific countermeasure to GNSS position spoofing is suggested in [331]. It verifies the GNSS position information by the help of extra radar based measurements. However, the proposed system is limited to verifying relative position information, as the reference sensors can only provide distance measurements. Thus, absolute position information as well as time information from GNSS cannot be validated. However, both data sets are required for VANET security functionalities [125, 176].

An approach to obtain location and time information in a secure way from a dedicated infrastructure is given by the *SecNav* protocol [258]. For VANETs, such an infrastructure could be provided by RSUs. However, this would afford to cover the whole road network by RSUs, which seems to be infeasible, due to economic constraints.

To limit the susceptibility of a VANET realization in regard to the found attack, multiple countermeasures are looked at in this work. Usage of more reference time sources than just GNSS in connection with mutual consistency checks is identified as a promising approach to limit the impact of a successful GNSS spoofing attack on a VANET.

A commonly used implementation for time synchronization, without connection to a backbone network, is given by the combination of well known tools GPS deamon (gpsd) for providing the GNSS time information and either NTP deamon (ntpd) or chrony [2, 3, 73], which adjust the local system time to a provided reference time. However, this solution does not support synchronization to multiple reference time sources. Thus, these implementations need to be extended to be used for the proposed countermeasures to attacks on GNSS time spoofing.

The following chapter introduces the evaluation methodologies and tools used in the remaining chapters of this work.

Chapter 3

Evaluation Methodology for VANET Security Mechanisms

This chapter describes the evaluation methodology used throughout this work. It is based on simulation of an ETSI ITS based VANET inside a dedicated simulation environment as well as on performance measurements of algorithms used in these networks, which are conducted on real hardware. A discrete time simulation approach is used in typical VANET simulators [215, 266, 277]. Hence, these simulators cannot be used to evaluate the runtime performance impact of applied algorithms inside nodes. Thus, we use performance measurements on real hardware to evaluate on this aspect.

Evaluation of VANET mechanisms heavily relies on simulations. This is caused by high effort of conducting real world field tests and the difficulty to realize complex traffic scenarios in a reproducible way during such field tests [288]. Hence, approaches proposed in this work have been implemented inside a simulation environment to obtain performance metrics for them. These metrics are used to to compare them to proposals from related work.

The further outline is as follows. Firstly, Section 3.1 introduces the applied performance metrics in regard to VANET protocol behavior. Afterwards, Section 3.2 treats the used traffic scenarios. Section 3.3 provides a description of the utilized simulation environment combining multiple dedicated simulators. Finally, methods for computational effort measurement are provided in Section 3.4.

3.1 Metrics

The main metrics for VANET performance used in this work are

- the certificate (chain) emission rate,
- the number of packets lost due to cryptographic packet loss, and
- channel load in terms of channel busy ratio (CHBR).

Both certificate emission rate as well as cryptographic packet loss are measured within the implementation of the VANET protocol stack. The certificate emission rate can relate to pure PSC

or AAC emission or the transmission of a certificate chain, which is in ETSI ITS the combined sending of PSC and AAC.

In contrast, the channel load is measured by equipping nodes within the network simulator with additional wireless devices at the same location as the devices conducting VANET communication. Such extra devices do never transmit, but only probe the wireless channel at each time stamp of the discrete event simulation environment (see also Section 3.3) to determine whether the channel is busy or not. Parameters of such extra receivers are identical to the ones used for ordinary VANET communication.

The caused channel load by an individual node (partly) depends on the average number of certificate inclusions in the security envelope per second. Attacks or cross influence from other VANET functionality may change the frequency of certificate emission. For such cases the relation between emission frequency in case of a present influence and an uninfluenced system is taken into regard.

For all metrics the standard deviation of obtained measurement results is determined and given in corresponding illustrations in later chapters of this work. Thereby, reliability and significance of the obtained results are illustrated. This also addresses criticism from [92] on the way of presentation of evaluation results in large parts of the VANET related literature.

The metric used in [135], which is called cooperative awareness, is not used within this work. The core reason is that this metric does not consider the data update rate at receivers. I.e,, it is only looked at how many nodes in a dedicated part of a node's environment use a PSC being known to the ego node, but it is not considered whether the ego node receives any message from these nodes at all. This is especially a problem in urban environments. In such scenarios, two nodes can be very close (e.g., closer than 100 m), but no message exchange is possible due to shadowing from a building. Such nodes have a negative influence on the cooperative awareness as defined in [133, 135], although there is never any message exchange between both nodes. Therefore, no cryptographic packet loss occurs and the presence of such nodes should not be counted as a negative performance criteria of the PSC distribution strategy.

The next section introduces the traffic scenarios, which are used for evaluations throughout this thesis.

3.2 Traffic Scenarios

An overview of popular traffic scenarios in prior work is given in Section 2.4.3. The following road network topologies are considered within this work.

- freeway scenario: three lanes in each direction and 6 km length (see Figure 3.4), e.g., used in [106, 190]
- rural road: a straight road with one lane in each direction, 6 km length and extra vehicles joining the road about in the center of the scenario, like suggested in [204] (see Figure 3.1). The joining roads have only one lane with traffic going towards the main road.
- urban grid: represents Munich Schwanthalerhöhe (see Figure 3.3), as exported from Open Street Map on 17th July 2014 [241]. Urban grid scenarios are used, e.g., in [288, 304].

- urban roundabout: represents the quite large roundabout to be found in Munich Maxvorstadt (see Figure 3.2), as exported from Open Street Map on 17th July 2014 [241]. Usage of this kind of scenario is suggested in [93].

Traffic on the road topologies freeway and rural road is determined by defining deterministic traffic flows within SUMO. In contrast, traffic for the much more complex topologies of urban grid and roundabout is created using the SUMO random trip generator[1].

Figure 3.1: Road topology of the rural road scenario.

Figure 3.2 shows the roundabout scenario (SUMO screen shot). The yellow triangles represent vehicles on the road. Random trips start and end at the edge of each road leading towards the roundabout, and each start position is connected with each possible end position of a trip.

The urban grid scenario is displayed in Figure 3.3. Comparison to the roundabout scenario (see Figure 3.2) shows that this scenario is more affected by shadowing. Both scenarios share low to medium mobility of nodes, due to applied speed limits according to urban environments.

In regard to velocity profiles a maximum velocity of 50 $\frac{km}{h}$ is set in the urban roundabout and the urban grid scenarios for all vehicles. The rural road scenario uses speed limits of 100 $\frac{km}{h}$ on the rural road itself and 50 $\frac{km}{h}$ on the roads being connected to the main road at about the center of the scenario. In the freeway scenario different traffic flows with individual maximum velocities are used on the dedicated lanes following the recommendations in [106]. The most right lane is preferred by vehicles using a maximum velocity of 80 $\frac{km}{h}$, vehicles on the middle lane yield 110 $\frac{km}{h}$, and the most left lane is preferred by vehicles going up to 130 $\frac{km}{h}$.

Parameters for traffic flows are derived from traffic densities. In doing so, the traffic density is varied from 16 to 45 $\frac{vehicles}{kilometer}$ [146]. The freeway scenarios include the ones recommended in [106]. To obtain results for VANETs realized within the given traffic scenarios, the simulation environment discussed in the next section is used.

[1] Implementation of the mentioned traffic scenarios was done in close cooperation with co-authors of [31, 33].

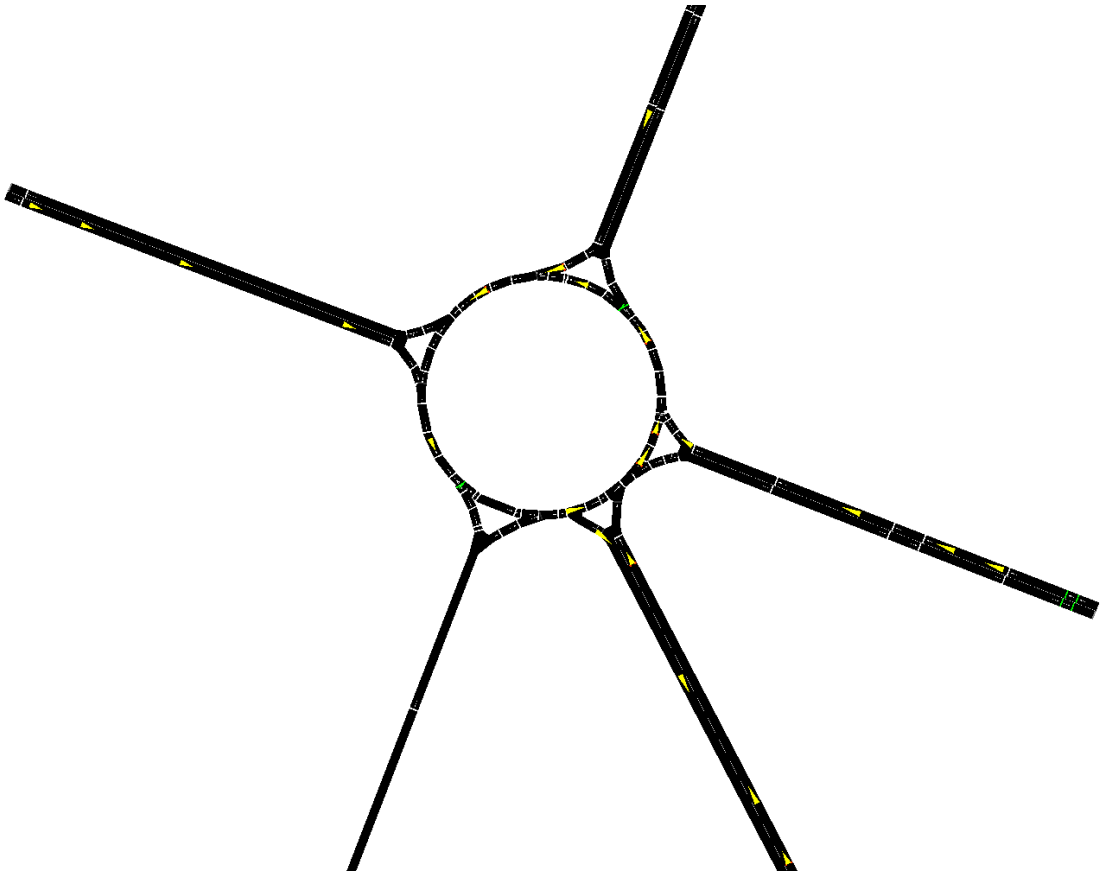


Figure 3.2: Road topology of the urban roundabout scenario.

3.3 Simulation Environment for VANET Security Mechanisms

An overview about commonly used simulation environments is provided in Section 2.4.1. Throughout this work a penetration rate of 100% is assumed, i.e., each vehicle on the road acts as a node of the VANET. All simulations are performed using the ezCar2X framework, which provides a full featured ETSI ITS protocol stack. Its correctness has been extensively tested, e.g., at ETSI's 3rd ITS Cooperative Mobility Services Plugtest [107] and against the independent implementation from [211,310]. For simulation purposes the ezCar2X protocol stack is embedded into ns-3, which is coupled with SUMO using the so called TraCI interface. Each node within ns-3 is equipped with its own protocol stack. The internal time of each node does not start with time stamp zero, to avoid unintended synchronization effects between the sending times of messages between nodes. Instead, a randomly generated time offset in the interval between zero and one second is used as the internal start-up time of each node.

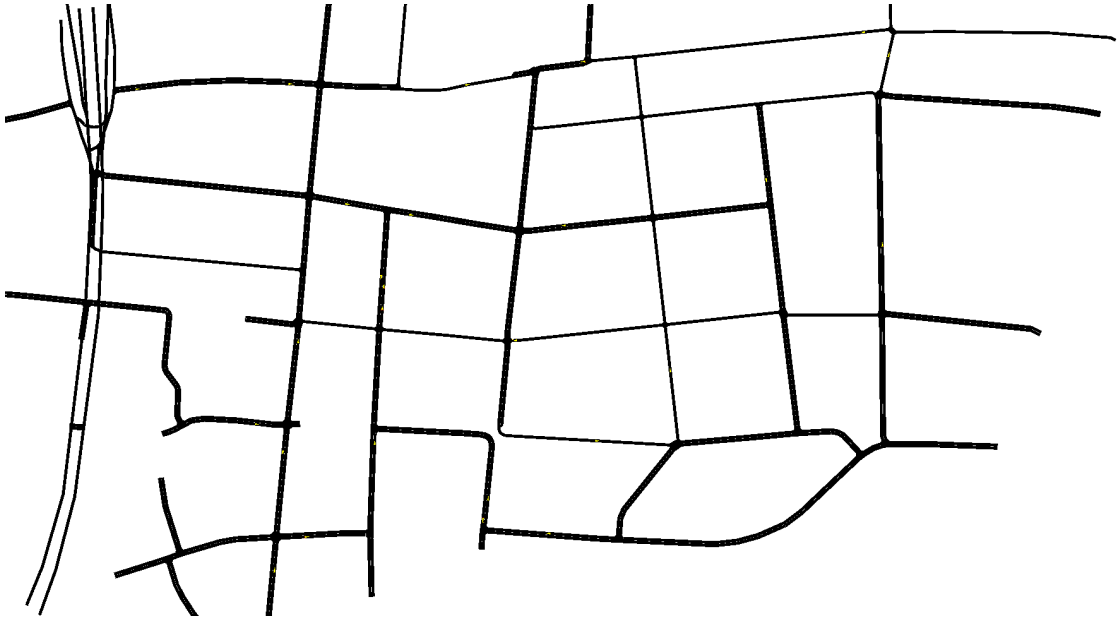


Figure 3.3: Road topology of the urban grid scenario.

For more details about ezCar2X and the simulator setup the reader is referred to [46, 266, 267, 285][2,3,4].

To determine the metrics from Section 3.1, the so called *core zone* concept is applied to all simulations. Considered metrics are only calculated from the measured results within the core zone, which is a subset of the full simulated area. The subset is chosen in a way to be surrounded by the extra simulated area avoiding edge effects like described in [106, 191]. Moreover, only results from vehicles having finished their journey through the entire core zone are collected. The core zone concept is illustrated in Figure 3.4 for the freeway scenario.

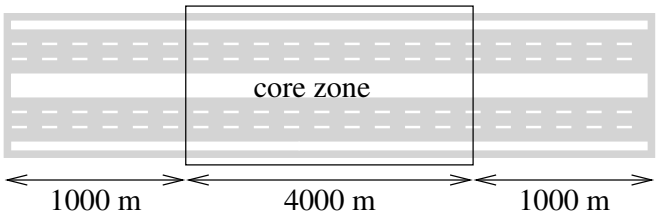


Figure 3.4: Core zone and road topology of the freeway scenario.

Collection of data for evaluation metrics is started after an initialization time has elapsed.

[2]The author's contribution to the ezCar2X framework mainly relates to the design and implementation of security functionality, platform independent data encoding schemes and integration into ns-3. Work on the ezCar2X framework was done in close cooperation with co-authors of [32, 36, 45, 46, 266, 285].

[3]For remarks on [266] see footnote 1.

[4]The author's contribution to [46, 285] is mainly to the proposed software engineering and evaluation methodologies as well as to the implementation of regarded ADAS. Remaining contributions are from the coauthors.

During this time the traffic flow on the chosen road topology builds up. This initialization period is excluded from the evaluation, as it does not model a scenario, which is likely to happen in practice. For each one of the used road topologies an individually adjusted initialization time is used. Moreover, the overall simulated time span is determined by defining the number of nodes, which need to have finished their journey through the core zone, before the simulation is stopped. A number of 3000 nodes is used for all traffic scenarios, except of the freeway scenario. To evaluate the freeway scenario a number of 9000 nodes has been used.

The cross-layer message size problem identified in Section 4.3 is avoided in simulations by not enforcing the DCC rule for maximum packet size on the access layer. Otherwise, no valid operation of the protocol stack could be achieved without applying changes to the facility or security related data sets. For more details about this issue see Section 4.3.

On the physical layer a pathloss model with Nakagami fading is used. Its parameters are taken from [64, 65], while the remaining implementation is as provided by ns-3 [263]. The individual parameter sets for freeway, rural road and urban environments are applied when simulating the corresponding traffic scenarios from Section 3.2. I.e., the highway and rural road scenarios are combined with corresponding channel model parameters from [65], while the channel model parameters from [64] are used for both urban scenarios. Parameters from [64, 65] are used, as they provide a consistent set of parameters for different traffic scenarios. There is no additional simulation of shadowing effects. General information about channel models for VANET communication is available in [200, 314].

3.3.1 Advantages of the Full Feature Protocol Support

Missing or outdated features on various protocol stack entities make it impossible to correctly evaluate their corresponding impact on overall system performance. Examples of issues of ETSI ITS, which have been found in this work using the framework outline above, but cannot be identified by incomplete evaluation environments, like iTETRIS, are given by,

- inability to send most of the to be distributed messages, due to maximum messages size violations (from DCC) on the MAC layer (see Section 4.3), and
- impossible encrypted multi-hop communication, due to encryption of data sets needed for forwarding (see Section 6.6).

Additionally, basing research on top of incomplete evaluation environments can lead to the proposal of inappropriate approaches. An example is duplicating information on various protocol layers, like in [57]. In that work, adding the so called ITS-Application Identifier (AID) (ITS_AID) to the GeoNetworking header is suggested. However, this data set is contained in the security envelope and the standardized interface between both parts of the protocol stack specifies to hand over this information [125]. Thus, presence of the ITS_AID in the network layer meta data is pure overhead, as this information is already provided by other mechanisms. Another example is given by the PSC depletion attack from [251], as explained in Section 2.3.2.

3.3.2 Comparison of Simulation Frameworks

Evaluation results for VANETs should be obtained in a way, which allows reasonable expectation of similar behavior in real world realizations. One part of accurate VANET modeling in simulation environments is to accurately implement communication stacks inside such environments. Table 3.1 gives an overview about MAC layer message sizes obtained from different simulation frameworks. All values are given in bytes. VSimRTI is not considered here, due to a lack of access to this commercial framework. This topic is also covered in the author's prior work given in [43][5].

	standardized	Veins	Artery	iTETRIS	ezCar2X
BSM	215	70	-	-	-
CAM	217	-	220	300	217

Table 3.1: MAC layer message size comparison of simulation frameworks.

To obtain the values given in Table 3.1, a CAM and BSM holding only mandatory data sets have been used. Hence, no optional container is present at the facility / application layer and no certificate is included in the security envelope. iTETRIS is used in version 0.3.0, Veins in version 4.4 and the tested checkout of the Artery repository is from the 1st June 2016. Considered standards are [103, 119, 122, 125, 174, 175, 271].

Deviations of figures in the first column (standardized message size) of Table 3.1 from those in other columns (message sizes from implementations) mean that standards have been implemented inaccurately. There are various reasons for the found deviations. For example, missing support for security functionality leading to a missing security envelope causes significantly shorter messages, especially for Veins (see also Table 2.1).

The size deviation between Artery and ezCar2X is caused by the usage of an older version of the ETSI ITS security envelope's standard in Artery. The minimum size security envelope has been shortened by three bytes during the change from [109] to [125]. Moreover, [109] does not support certificate chain distribution, in contrast to [125]. Hence, this mechanism cannot be studied by using Artery.

In Veins the message size is found to be significantly smaller than the size obtained by inspection of relevant WAVE standards. An in-detail review of the Veins's source code shows that this framework does not use any kind of VANET specific data representation. Instead, the generic message data representation format of OMNET++ is used. The message size depends on the data types of contained data fields. Moreover, no security envelope is implemented, which accounts for the biggest share of missing message size (96 bytes). The obtained message size deviation can be expected to lower the channel load to about one third in comparison to the one experienced by using the standardized approach. Clearly, an impact on evaluation results for applications can be expected from such a difference.

Veins has been used to evaluate the performance of security mechanisms of WAVE, e.g., in [92, 93]. However, the corresponding extensions of the framework are not publicly available. Hence, they are not considered here.

[5] See also footnote 2.

Within the iTETRIS framework the message size is specified within a configuration file for all nodes. A default value of 300 bytes is used in the considered version of the framework. The message size is completely independent of the content, which gets disseminated by nodes. This only works because the actual content of messages is not sent over the simulated wireless connection within ns-3. Instead, ns-3 is only used to determine whether the message is received by a particular node, while the real content of messages gets exchanged via a second, independent mechanism. This means that the impact of average message size reduction schemes, like sporadic inclusion of data sets, e.g., PSCs, on VANET functionalities cannot be evaluated with iTETRIS, as it always uses a fixed message size. In contrast, this is possible using all other considered frameworks. Furthermore, inspection of the source code of the iTETRIS framework shows that the implemented standards' versions are greatly outdated.

The obtained results show that the ezCar2X framework accurately resembles the currently standardized ETSI ITS system. Other considered frameworks show drawbacks in regard to accurate VANET protocol modeling as outlined above. Hence, the ezCar2X framework is used for further evaluations throughout this work.

3.4 Computational Effort Measurement

Basic mechanisms of computational effort measurement are introduced in Section 2.4.2. In regard to computational effort two metrics are considered throughout this work. These are

- runtime, and
- memory requirements in regard to stack and heap consumption.

To obtain reliable results regarding the runtime of an algorithm, the corresponding test program is run on an otherwise idle system. The operating system is a standard Debian Linux installation. All test programs are implemented using C++. Instructions from [246] are applied to avoid unintended influence of out-of-order instruction execution of modern processors on the measurements.

For all timing measurements, the Linux kernel's high performance counters are utilized. These can be accessed from user space by calling the clock_gettime() function [177]. In doing so, CLOCK_PROCESS_CPUTIME_ID is used as the clock ID to determine only the time spent in the process containing the to be evaluated algorithm. An accuracy of up to 1 ns can be achieved, in case the underlying hardware permits such accurate measurements [183]. The described methodology for time measurements is preferred over directly reading a processor's time stamp counter (TSC), as used in other work, e.g., in [246]. This is done as [246] uses operations only available inside the Linux kernel itself. However, the measurements within the conducted performance studies are done in user space. Thus, some prerequisites of the approach from [246], e.g., disabling of interrupts and scheduling, cannot be fulfilled. Therefore, the performed measurements rely on the implementation of the clock counter in the Linux kernel.

Runtime measurements of algorithms are repeated 10.000 times, and the average of the obtained results is calculated. To evaluate the reliability of obtained results, their standard deviation [131] is determined.

All test programs are compiled on the target using the GNU Compiler Collection (GCC) in version 4.8.2 [257]. Strong optimization is enabled with the *-O3* compiler flag.

Three different processor technologies are used for runtime measurements in this work. These are a AMD Geode, an Intel Atom and an Intel Core i7 processor [9,170,171]. An overview about their individual characteristics is given in Table 3.2.

type	AMD Geode LX	Intel Atom Z520PT	Intel Core i7-2640M
clock speed	500 MHz	1.33 GHz	2.8 GHz
measurement resolution	2 ns	1 ns	1 ns

Table 3.2: Used processors and achievable measurement accuracy via Linux clock counters.

An algorithm's main memory footprint (heap and stack utilization) can be measured by usage of the so called *malloc_count* framework [22]. This framework allows to trace the memory behavior of arbitrary parts of a program by inserting dedicated function calls into it, i.e., by manual instrumentation. Such extra function calls were only used for memory measurements, i.e,, they were removed during timing measurements, as they introduce additional runtime overhead. Other memory tracing tools like *massiv* from the *valgrind* framework do not allow to adjust measurement procedures with such fine granularity [284]. Hence, malloc_count was used to obtain the results presented within this work.

The following chapter studies security related overhead in VANETs. In doing so, the evaluation methods discussed in this chapter are used.

Chapter 4

Security-related Overhead in VANETs

This chapter takes an in-detail look at various sources of overhead in VANETs in regard to security and privacy mechanisms. Such overhead limits the efficiency of VANET communication. To improve its efficiency, the different sources of overhead are categorized and analyzed to show their individual characteristics, and to identify possibilities to limit the amount of overhead. The given work is partly covered by prior work of the author in [44][1].

There are several sources of overhead in VANETs, which are caused by security mechanisms. Their influence on the overall system performance includes to cause an

1. increase in required bandwidth and restrict data size for higher protocol layers by the chosen
   (a) type of platform independent data representation influencing the encoded size of the envelope (see also Section 4.1 and point 2e),
   (b) inclusion of extra data sets, e.g., on demand included certificates [133, 135, 185, 191, 275], and the used
   (c) digital signature algorithm [279],
   (d) certificate type (explicit or implicit) [53, 125, 176],
2. extra data reception delay at receivers built up by
   (a) channel access delays at senders (rises with channel load / nodes' data rate needs) due to carrier sense multiple access - collision avoidance (CSMA-CA),
   (b) pure transmission time (rises with message size),
   (c) signing delay from creating the digital signature at the message's sender,
   (d) authentication delay either from signature verification [279] or from discarded packets due to missing security parameters (see also point 3),
   (e) platform independent data representation (sometimes called data serialization) affecting processing time at sender and receiver (see also Section 4.1 and point 1a),

[1] Contribution of co-authors mainly relates to the determination of data size requirements from protocol layers except of the security functionality. The main contribution is from the author of this work.

3. cryptographic packet loss (for CAMs see, e.g., [135]),

4. storage space increase [134], and

5. pseudonym changes.

The delay caused by message signing is typically considered of minor influence in comparison to signature verification delay. High numbers of received messages in short time spans require high computational performance for verification of all incoming messages (so called verify-all). In contrast, the number of messages sent by an individual node is usually low, e.g., ten CAMs per second. A popular method to limit the verification load is to verify only a subset of all received messages, e.g., by VoD [199], as introduced in Section 2.2.4.6. However, we show in Section 5.2 that this method endangers system reliability and robustness, e.g., by creating a DOS vulnerability.

For the case of cryptographic packet loss (point 3), related work has only studied message discarding due to missing knowledge of the corresponding PSC. Thus, only the loss of a CAM or BSM has been considered, as messages with remaining security profiles (e.g., DENMs) always include the PSC [125]. However, the multi level certificate hierarchy used in ETSI ITS and WAVE may also cause an inability to verify a message due to missing knowledge of higher level certificates than the lowest level certificate, i.e., the PSC. This affects especially multi-hop messages with a dissemination area superseding the one of CAMs sent by the same originator. Only CAMs are used to distribute AACs [125]. Thus, nodes outside the distribution area of CAMs emerging from the sender of a DENM cannot obtain the AAC used to secure the DENM from its sender. Moreover, other messages than CAMs (or BSMs in WAVE) cannot cause a request for a missing AAC. Hence, verification of such messages completely relies on prior distribution of the full certificate chain by periodic beacon messages.

The following section provides a comparison of platform independent data representation schemes applied to the ETSI ITS security envelope. Thereby, the aim is to identify possibilities to minimize the encoded data length, while keeping the computational effort for data encoding and decoding low (see also points 1a and 2e above).

4.1 Platform Independent Data Representation

Platform independent data representation schemes are commonly used for encoding data to be sent from one node to another one within a network. Selection criteria for a data representation scheme include the length of the encoded data as well as computational performance of the encoding and decoding procedures. In highly bandwidth restricted systems, like VANETs, especially the encoded data length is a key selection criterion. However, no comparison of data representation schemes applied to the ETSI ITS security envelope has been published in prior work. Hence, such kind of evaluation is provided in this section. The content provided in the following is partly covered by prior work of the author published in [36, 38][2].

[2]Contribution of co-authors mainly relates to the implementation of the used EXI encoding scheme as well as to data representation within CAM and DENM data structures (not covered in this work). The main contribution is from the author of this work.

Within VANET protocol stacks two different data representation types are used. These are custom binary encoding schemes and Abstract Syntax Notation 1 (ASN.1) based encoding. An overview about their usage on different protocol layers within ETSI ITS and WAVE is given in Table 4.1.

layer	ETSI ITS	WAVE
application / facility	ASN.1 UPER	binary
transport	binary	binary
network	binary	ASN.1 DER
security	binary	ASN.1 DER
MAC / PHY	binary	binary

Table 4.1: Data representation schemes used in VANET standards.

As one can see from Table 4.1, ASN.1 is used in different variants Data Encoding Rules (DER) and UPER. UPER provides a more compact, i.e., shorter data representation in comparison to DER [90]. To keep the security overhead low in regard to message size increase, the data representation used for the security envelope should be chosen to provide an encoded length as short as possible. Hence, a comparison of the security envelope's size in case of binary and ASN.1 UPER encoding is provided in the following. ASN.1 UPER encoding for the security envelope within ETSI ITS is proposed in [110].

Other popular schemes for platform independent data representation, except from ASN.1, include Google Protocol Buffers (protobuf) and XML-based approaches like Efficient XML Interchange (EXI) [149, 150, 311]. protobuf is often regarded as being able to outperform other data serialization schemes in regard to required computational resources [143]. Moreover, EXI is a promising approach to achieve a compact, i.e., bandwidth saving, data representation for wireless communication [51]. Thus, protobuf and EXI are also regarded in the provided comparison of data representation schemes for the security envelope's data sets of ETSI ITS. An extension of this study to facility layer's data sets of CAM and DENM can be found in [38].

The used C++ implementations for performance tests use dedicated libraries for the ASN.1, protobuf and EXI related functionalities. These are FFASN1 [19] for ASN.1 UPER, libprotobuf in version 2.5.0 as provided by Google and Embeddable EXI Processor in C (exip) for EXI [201]. Moreover, correct ASN.1 encoding was cross checked with software from OSS Nokalva [243]. All implementations of the four different data representation schemes were conducted within the ezCar2X framework. This framework also provides the binary data representation scheme. For the performed test runs, a common data set for the to be encoded values was used.

ASN.1, protobuf and EXI use a dedicated format to specify the content of a data set, which should be represented in a platform independent way. Typically, these specifications are used by a specific interpreter tool to generate code. This code performs the use case specific tasks of the data encoding (i.e., raw data to encoded data) and decoding (i.e., encoded data to raw data) procedures. This holds for the used FFASN1 and protobuf implementations. In contrast, the exip library uses auto-generated code only for the decoding procedure. All EXI encoding tasks are performed using a general purpose encoder on an XML representation of the to be encoded data. In case of EXI, data structures are described using well known XML schema files.

The definition files for protobuf and the XML schema files for EXI were derived from the ASN.1 definitions given in [110]. Transformation from ASN.1 definitions to protobuf and EXI is straightforward, due to the low number of available data types in both data representation schemes. During the transformation process always the smallest protobuf (or EXI) data type, which is able to hold the corresponding ASN.1 data type, was selected to avoid introducing unnecessary overhead.

Protobuf does not provide a data element for choices, thus all possible subjects of a choice where chosen to be optional elements. This also means that, the protobuf library does not provide any possibility to check whether exactly one of the to be chosen elements was actually chosen. Thus, this check is left to the user of the auto-generated code.

In the performance study case for EXI, two approaches were followed. At first, a full mapping of the standard to an EXI schema has been developed. These schema files are found to contain a lot of nesting levels, leading often to (informationally) unnecessary content, which gets explicitly encoded in the serialized data representation [311]. This makes such schemes become easy to expand and very well structured. However, since one of the key parameters in this study is the size of the encoded messages, extra data optimized schemes are designed. In the data size optimized schema files the unnecessary nesting levels are merged, which decreases the number of options getting encoded during EXI based data serialization. An example for the difference in respect to the XML structure of elements is given in Listings 4.1 and 4.2 for the case of the so called VerificationKey data structure from the security envelope [109].

```
<s0:VerificationKey>
  <s0:Key>
    <s0:EcdsaNistp256WithSha256>
      <s0:publicKey>
          <s0:CompressedLsbY0>
            <s0:x>FFFFFFFF</s0:x>
        </s0:CompressedLsbY0>
      </s0:publicKey>
    </s0:EcdsaNistp256WithSha256>
  </s0:Key>
</s0:VerificationKey>
```

Listing 4.1: VerificationKey element from the security envelope as implemented according to the standard.

```
<s0:SubjectAttributeVerificationKeyEcdsaNistp256WithSha256
  CompressedLsbY0>
   FFFFFFFF
</s0:SubjectAttributeVerificationKeyEcdsaNistp256WithSha256
  CompressedLsbY0>
```

Listing 4.2: Data size optimized VerificationKey element from the security envelope using a dedicated XML tag instead of a deep hierarchy of tags.

As one can see from comparing Listings 4.1 and 4.2, the number of tags required for storing

the same amount of payload is reduced from six to only one. This significantly reduces the amount of meta data stored in the serialized data, which leads to reduced encoded data length. The optimized XML structure is similar to the one of the binary encoding scheme, which also lacks encoding of data structure hierarchies [109].

For the cases of ASN.1, protobuf and EXI the encoded size of data sets may depend on the encoded content, e.g., due to variable length integer encoding [90, 311]. To obtain realistic measurement results, a randomly taken time stamp and a valid certificate, e.g., with correct ECC parameters, are used during all measurements. For both the time stamp and the certificate 100 different data sets were randomly generated and the encoding procedure was run with all considered encoding schemes, to check for a difference in encoded data length. The results showed an equal encoding length for all tested data sets. Hence, the same example data set is used for all measurements discussed in the following sections.

Evaluations in regard to data length as well as computational effort required for encoding and decoding in case of the different data representation schemes (binary, ASN.1, protobuf, EXI) are given in the following.

4.1.1 Data Size Requirements

The obtained results for encoded data lengths of the ETSI ITS security envelope in regard to binary, ASN.1, protobuf, and EXI representation are given in Table 4.2. The security envelope for CAMs occurs twice, as it has two variants, which are with and without an included PSC (see also Section 2.2.4.3). Certificate chain inclusion in CAMs is not considered in this case, as results in Section 6.3 show that distribution of just one certificate per CAM is sufficient. All values are given in bytes.

profile	binary	protobuf	ASN.1	EXI
1 (CAM) w/o cert.	96	133	88	90 (opt: 87)
1 (CAM) with PSC	222	306	240	210 (opt: 201)
2 (DENM)	233	318	249	215 (opt: 206)
3 (generic)	230	312	247	213 (opt: 204)

Table 4.2: Performance results in regard to encoded data length for encoding the security envelope with several data serialization schemes. All values are given in bytes.

One can see from Table 4.2 that protobuf encoding significantly increases the size of the security envelope in comparison to all other considered schemes. The encoding lengths for security profiles two (DENM) and three (generic) would not differ in case of ASN.1 encoding, as the data field called *message type* is optional according to [109], but required according to the ASN.1 definition from [110]. The presence of this data field is the only difference between these two security profiles. Thus, this difference would vanish in the case of ASN.1 encoding, i.e., both cases would yield an encoded data length of 249 bytes for the security envelope. To justify the existence of both security profiles, the used protobuf and EXI definitions declare the message type field as optional. The ASN.1 definition is also changed in regard to this point. From a semantic point of view, it makes no sense to give a message type in case of profile

number three, as this is the default profile for messages of type *generic*, i.e., messages with no assigned message type.

Results in Table 4.2 show that in all cases binary encoding clearly outperforms protobuf in respect to achieved encoding length. Furthermore, it outperforms ASN.1 encoding in three out of four cases. The only exception is the case of security profile number 1 without PSC. In this case, ASN.1 encoding requires nine bytes less than binary encoding. However, for the case with certificate and security profile one, ASN.1 requires 19 more bytes than binary encoding. Moreover, binary encoding requires 18 bytes less for security profile number two and 21 bytes less for security profile number 3, respectively.

Results from Table 4.2 also show that the straight forward EXI encoding scheme achieves the smallest packet size for security profile one with certificate as well as profiles two and three. Furthermore, for the case of security profile one without certificate it is only slightly outperformed by the ASN.1 encoding scheme. However, the optimized variant of EXI encoding significantly outperforms all other schemes in regard to message size.

To give more insight on the best performing encoding scheme for security profile one, the average size of the security envelope should be considered. Due to the varying CAM emission frequency f_{CAM} (1 - 10 Hz, $f_{CAM} \in \mathbb{N}$) and the various certificate inclusion rules (see Section 2.2.4.3 or [109]), only a lower limit for the average size of the security envelope for profile one can be given. The average size of the security envelope $\overline{s}_{sec}$ is given by

$$\overline{s}_{sec} = \frac{(f_{CAM} - f_{cert}) \cdot s_{w/o} + f_{cert} \cdot s_w}{f_{CAM}}; f_{cert} \leq f_{CAM}. \tag{4.1}$$

The size of the security envelope without an included certificate is denoted by $s_{w/o}$, and the one with included certificate by s_w. $f_{cert} \in \mathbb{N}$ represents the certificate inclusion frequency. To calculate the lower limit of $\overline{s}_{sec}$, the maximum CAM emission frequency $\max(f_{CAM}) = 10$ Hz, and the minimum PSC inclusion frequency $\min(f_{cert}) = 1$ Hz is used. The results on $\min(\overline{s}_{sec})$ for the regarded encoding schemes are given in Table 4.3.

encoding scheme	binary	protobuf	ASN.1	EXI
$\min(\overline{s}_{sec})$ in bytes	108.6	150.3	103.2	102 (opt: 98.4)

Table 4.3: Minimum average size of the security envelope for CAMs (i.e., security profile one).

One can see from the results given in Table 4.3 that EXI encoding achieves the lowest result for $\min(\overline{s}_{sec})$. Moreover, $s_{w,EXI} < s_{w,\{ASN.1,protobuf,EXI\}}$ holds (see Table 4.2). This means that for a higher share of messages including a certificate the advantage of EXI over the other schemes is greater than in case of the minimum share of such messages leading to the results from Table 4.3. Hence, with EXI encoding the average message size will always be smaller than the one of other encoding schemes, whatever f_{CAM} and f_{cert} are applied. The maximum gain of EXI encoding over all other schemes is achieved in case $f_{cert} = f_{CAM}$ holds, i.e., every CAM's security envelope contains a certificate. In this case $\overline{s}_{sec} = s_w$ holds, and figures from Table 4.2 clearly show the data size reduction by using EXI encoding.

Regarding results from Table 4.3, the achieved message size reduction for the optimized variant of EXI in comparison to the straight forward EXI variant is an additional 3.53%. In

comparison to the standardized binary encoding scheme, it even saves 9.39% in message size.

Restructuring of the security envelope has been performed from standard versions [109] to [125] to, among other improvements, save just three bytes in data size. Hence, the found possibility to reduce the average size of the security envelope by EXI encoding by more than 10 bytes can be considered significant.

In the following, the performance of the different data representation schemes gets evaluated in terms of required runtime for encoding and decoding. To obtain reliable results and to make sure that results are not biased by the selected processor topology we use three different processors in the following. These are described in detail in Section 3.4. Obtained results are presented in the next sections.

4.1.2 Data Encoding Performance

Measured runtime performances for data encoding (i.e., serialization) for the security envelope of ETSI ITS are given in Figure 4.1. To obtain the given results, the evaluation methodology described in Section 3.4 is used together with implementation of the different data representation schemes as described above. The given averages are calculated from the results of 10.000 test runs.

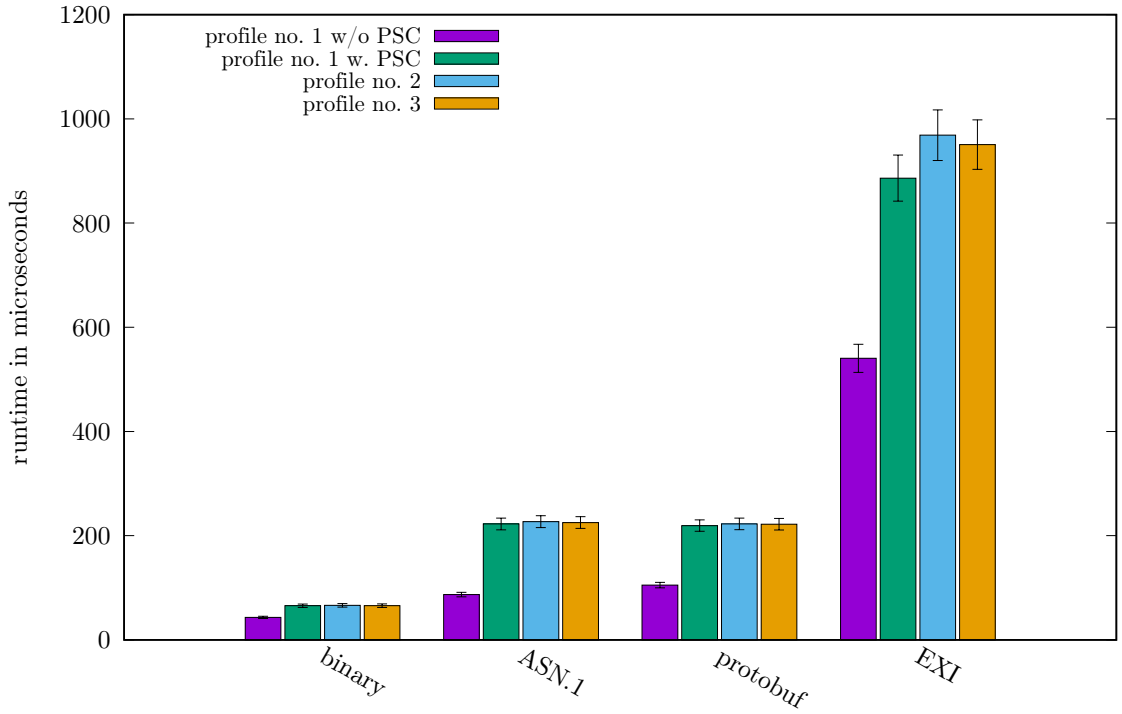
One can see from the results presented in Figure 4.1 that the binary scheme massively outperforms all other generic data representation schemes. Moreover, ASN.1 encoding outperforms its protobuf and EXI counterparts.

A significant source of influence on runtime performance for encoding the security envelope is the high number of small and deeply nested data fields used for defining the security envelope (see also Table 4.4). The achieved results depicted in Figure 4.1 indicate that binary encoding can handle this kind of structure better than the other encoding schemes. Moreover, ASN.1 and protobuf are almost on par and both clearly outperform the EXI mechanism.

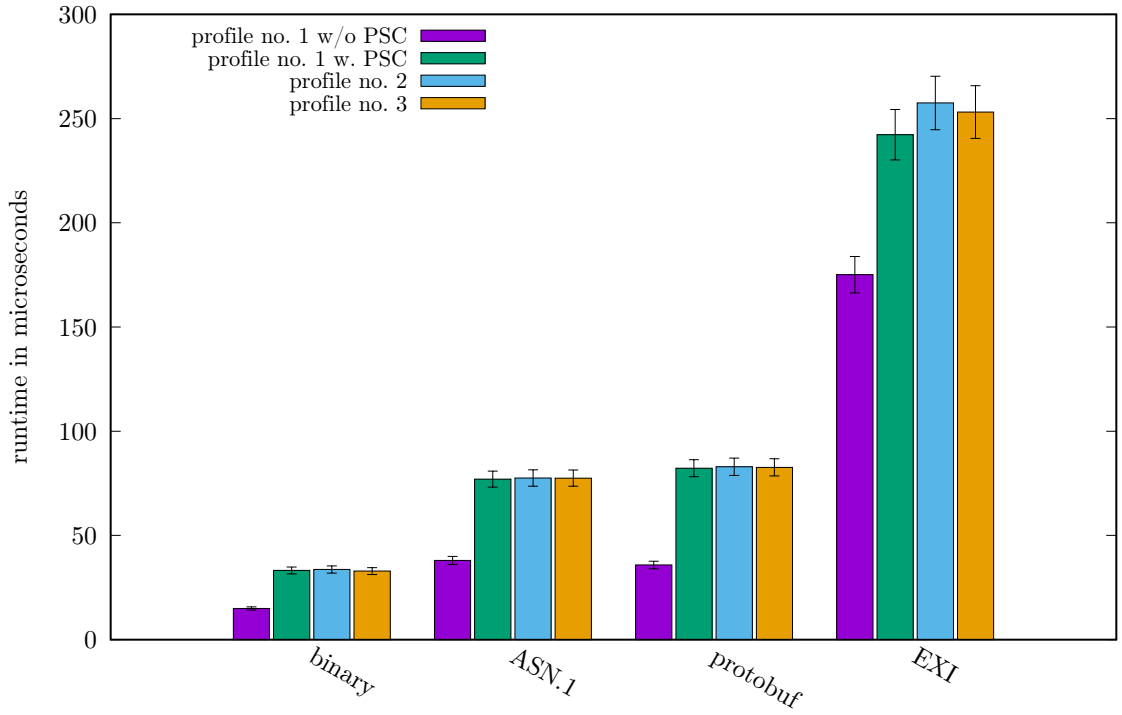
nesting level	1	2	3	4
sec. profile 1 (CAM) w/o cert.	5	16	4	0
sec. profile 1 (CAM) with PSC	5	22	21	11
sec. profile 2 (DENM)	5	22	23	11
sec. profile 3 (Generic)	5	22	22	11

Table 4.4: Nesting of data fields for individual security profiles' security envelopes.

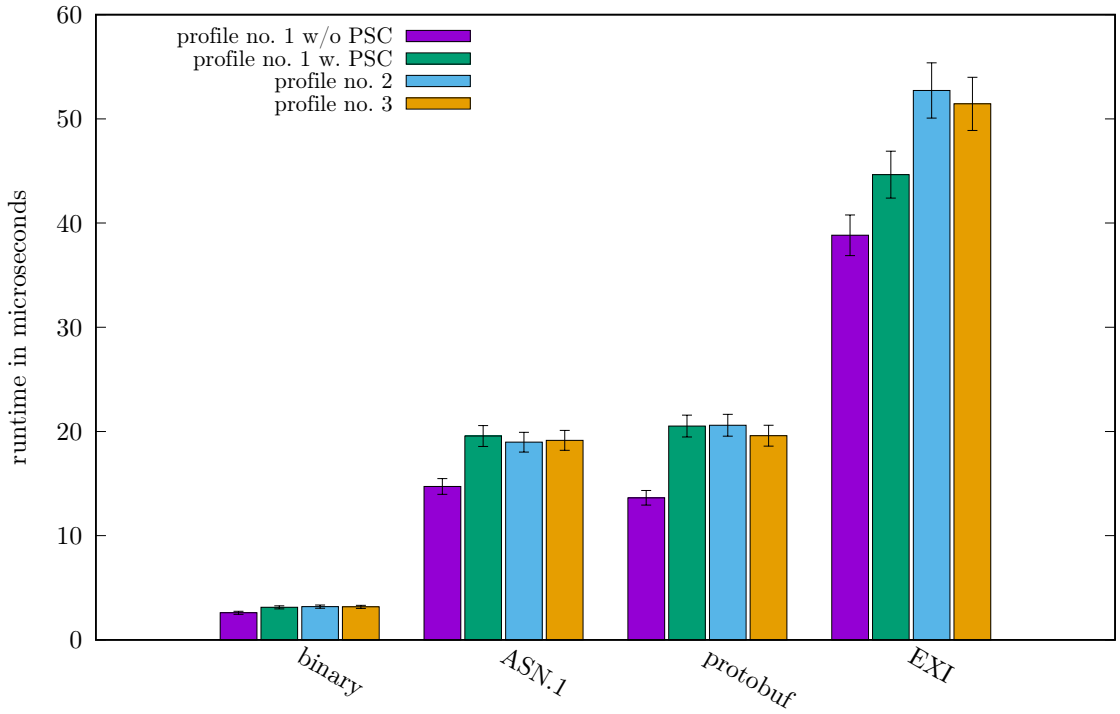
The numbers in Table 4.4 give the amount of data sets (mandatory and optional) found at the different nesting levels. To obtain the figures in Table 4.4, the full data sets were represented in a tree structure. As only mandatory data fields are used in the provided performance study, the elements of sub-trees following an optional element are not counted. Nesting level one means the top level of the data packet, whereas nesting level four relates to the data elements at the most deeply nested position inside the data packet. Such deep nesting is only present within the PSC's data structure. One can see from Figure 4.1 that for all data representation schemes the data structure with the least complex structure (security profile 1 without included certificate,



(a) Encoding runtime performance on AMD Geode processor.



(b) Encoding runtime performance on Intel Atom processor.



(c) Encoding runtime performance on Intel Core I7 processor.

Figure 4.1: Data encoding runtime performances on different processors.

see Table 4.4) requires significantly less computational performance in comparison to all other data structures.

For all results the standard deviation is given in Figure 4.1. In general, values for this statistical metric are small in comparison to the given averages. Therefore, the achieved measurement results can be regarded as significant. The differences between the obtained results for different encoding schemes for same encoded data content are much bigger than three times the standard deviation of the corresponding runtimes in almost all cases. Only the quite similar results for ASN.1 and protobuf show a significant overlap of their standard deviation intervals.

Comparing Figures 4.1c, 4.1b, and 4.1a one can see that except of a general increase in runtime (please note the different scaling of the vertical axis of figures), the overall results are the same for all processor technologies. This means the overall outcome of the performance study does not change by switching from a modern high speed processor (like the Core i7) to a quite old and low speed processor, like the AMD Geode. Lower processor speeds (see also Table 3.2) lead to increased runtimes, as can be expected. However, the increase is somewhat bigger than what can be calculated just by comparison of corresponding processor clock speeds. It is reasonable to observe an advantage in the runtime performance of the Core i7, which is due to the improved processor technology such as pre-caching algorithms, as it was introduced to the market significantly later than the used Atom and Geode processors.

Table 4.5 gives the results for main memory consumption for the ETSI ITS security enve-

lope's variants. In each cell, the first figure relates to the number of bytes consumed on the heap, while the second one gives the number of bytes used on the stack. Both values give the corresponding peak values during runtime of the measured algorithm. The profile column gives the number of the security profile as defined in [109]. As described before, the two cases of an envelope with and without certificate are to be distinguished for security profile number one (used for CAMs).

profile	binary	protobuf	ASN.1	EXI (optimized)
1 (CAM) w/o cert.	240 / 12168	1784 / 13528	1463 / 19784	61760 / 680
1 (CAM) with PSC	798 / 15800	3819 / 15016	2186 / 20528	63313 / 680
2 (DENM)	798 / 15800	4023 / 15016	2186 / 20528	63553 / 680
3 (generic)	798 / 15800	3865 / 15016	2186 / 20528	63457 / 680

Table 4.5: Encoding performance results for the security envelope. All values are given in bytes.

In regard to the sum of consumed memory (heap plus stack), binary encoding outperforms all other encoding schemes according to the results from Table 4.5. One should note that the high amount of consumed memory for the EXI implementation is due to the fact that the used encoder does not use any kind of a-priori code generation, as it is used for common ASN.1 and protobuf approaches. Instead the exip library [201] generates the encoding tree on-demand in memory, which is a highly flexible approach. However, it is targeted for experiments and not for mass roll out in embedded systems. Hence, practical realizations for VANETs should use another implementation approach with less demanding runtime and memory requirements.

4.1.3 Data Decoding Performance

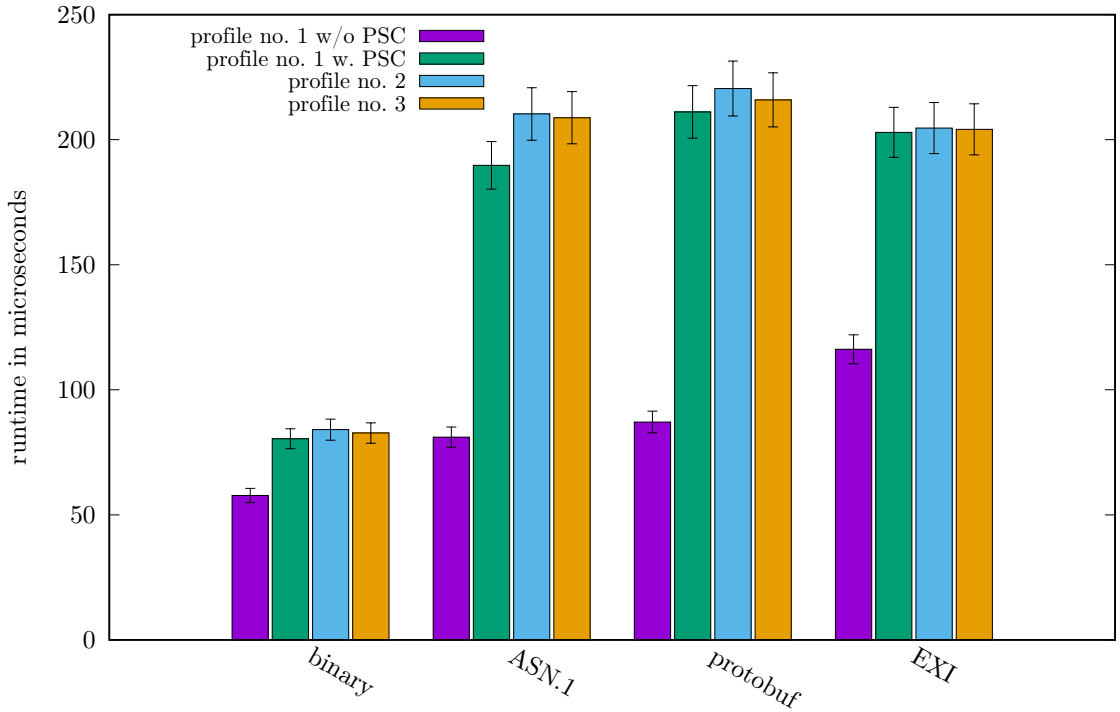
Runtime performance of data decoding (i.e., deserialization) is shown in Figure 4.2. Like in Section 4.1.2, the evaluation methodology from Section 3.4 is used to obtain all given results.

Data decoding is required at the receiver side of a message exchange to obtain the data, which was encoded by the sender. Typically, the number of received messages greatly exceeds the number of sent messages in a VANET, due to broadcast communication. Hence, efficient decoding is even more significant than efficient encoding. This is similar to the relation between message signing and verification.

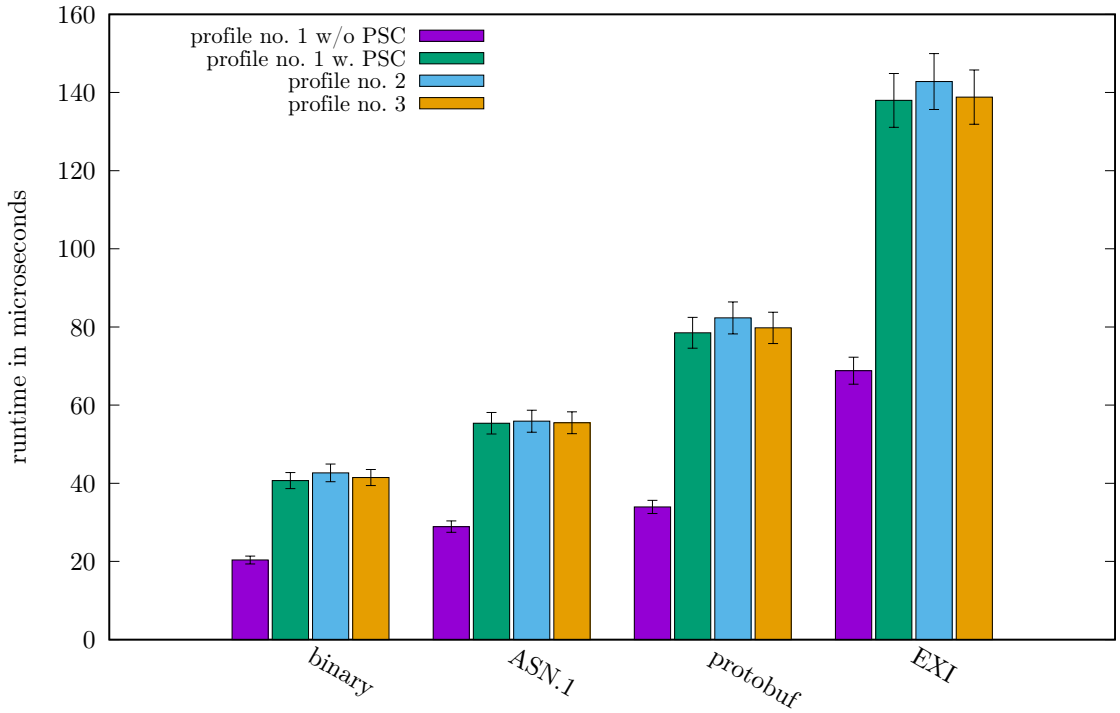
In general, the relation between the different encoding schemes in regard to their runtime performance is similar to the one obtained for encoding performance (Figure 4.2 vs. 4.1). Again, the binary scheme significantly outperforms the other generic data representation schemes. The EXI scheme performs worst, but the difference to protobuf is smaller for decoding then for encoding.

Table 4.6 summarizes the results for memory usage of the different deserialization schemes for the security envelope. Like in Table 4.5, the first value in each cell relates to peak heap memory consumption, while the second figure gives the corresponding result for stack utilization. All values are given in bytes.

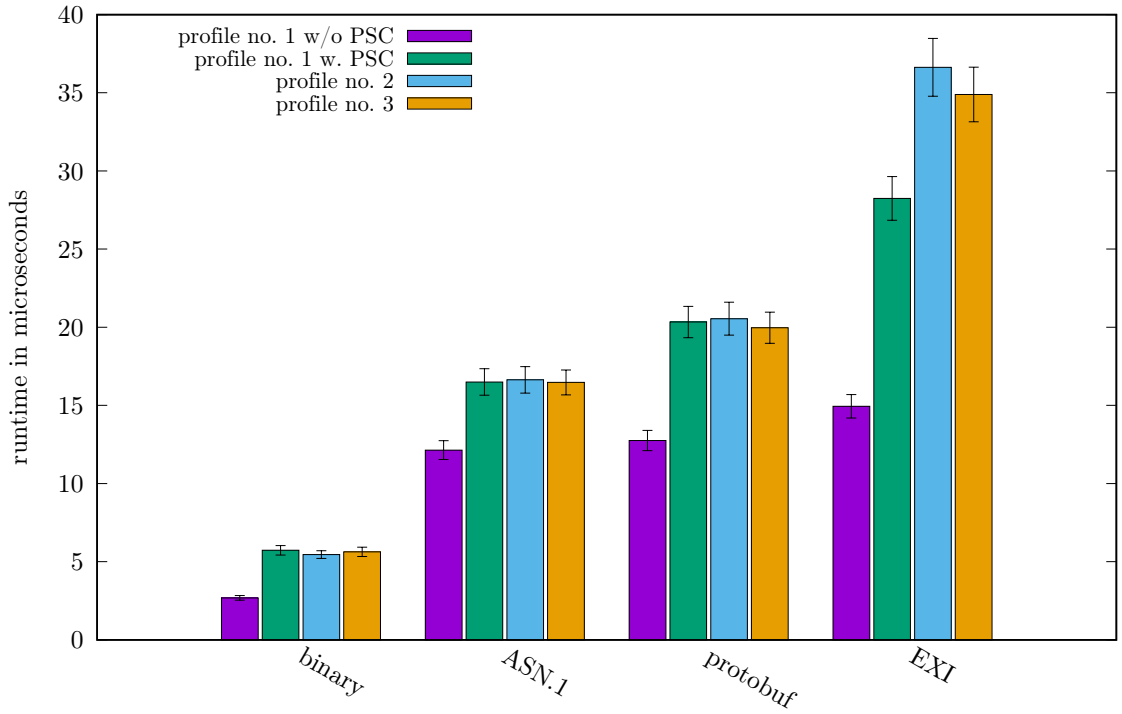
For decoding of the security envelope, ASN.1 uses less memory than protobuf. Moreover, EXI exhibits the smallest memory footprint for all security profiles, significantly outper-



(a) Decoding runtime performance on AMD Geode processor.



(b) Decoding runtime performance on Intel Atom processor.



(c) Decoding runtime performance on Intel Core I7 processor.

Figure 4.2: Data decoding runtime performances on different processors.

profile	binary	protobuf	ASN.1	EXI
1 (CAM) w/o cert.	872 / 15480	1916 / 19992	1296 / 13016	13375 / 1080
1 (CAM) with PSC	1709 / 19208	3665 / 20632	4255 / 14040	14131 / 1100
2 (DENM)	1773 / 19208	3869 / 20632	4327 / 14040	14195 / 1140
3 (generic)	1717 / 19208	3711 / 20632	4311 / 14040	14198 / 1136

Table 4.6: Decoding performance results for the security envelope. All values are given in bytes.

forming binary and ASN.1 decoding schemes. This clearly shows the dependence of a data (de-)serialization scheme on the used structure of the message.

Memory usage of EXI decoding is much smaller compared to encoding (see also Table 4.5). This is because the decoder design of the exip library does not try to build a full message tree in memory before returning the decoded message to the user. Instead, the approach is more like the one for simple binary decoding. The data packet is parsed element by element and for each primitive data type found (e.g., an integer) an a-priori registered callback function (provided by the user) is called. This usage of a-priori information clearly reduces memory consumption inside the decoding method.

From the obtained results for the quite different processor technologies, one can conclude that the achieved results can be used to interpret the behavior of the studied encoding algorithms

within embedded systems using medium to high speed processors.

Corresponding results for CAMs and DENMs encoded with ASN.1, protobuf, and EXI can be found in [38]. It is shown that ASN.1 UPER greatly outperforms protobuf in regard to runtime and memory requirements for ETSI ITS facility layer data sets within CAMs and DENMs.

4.1.4 Conclusion of Comparison

The used data representation approach is shown to have a significant impact on the overhead caused by security functionality in an ETSI ITS VANET. The comparison of binary, Google Protocol Buffers (protobuf), ASN.1 and EXI data representation schemes shows that the data size optimized EXI scheme provides the shortest size platform independent data representation. More than 9%, i.e., 10 bytes at least, in size can be save in average for the security envelope, in comparison to the standardized binary encoding scheme.

The provided evaluation of computational effort from Sections 4.1.2 and 4.1.3 clearly shows that binary data representation outperforms all of the considered general purpose representation schemes. Hence, a move from binary data representation towards ASN.1, like considered in [110] for ETSI ITS, is discouraged. This also holds for WAVE [175, 176], due to very high similarity of used security envelopes in ETSI ITS and WAVE. Instead, a move towards a data size optimized EXI variant is recommended. However, a more computationally efficient implementation than the considered general purpose EXI library is required for mass roll-out.

4.2 Certificate Distribution

The combination of requirements of verifying each received message and only sporadic certificate emission leads to cryptographic packet loss. This occurs in scenarios in which a node receives a message while the corresponding certificates are not available. Thus, the message cannot be verified. Hence, the receiver discards the message, i.e., it is lost.

The usage of a multi-level certificate hierarchy in ETSI ITS and WAVE leads to two distinct causes of cryptographic packet loss. Either a node's individual PSC is not available, or the certificate of a CA within the certificate chain of a PSC is unknown to the receiver. The first case is looked at in Section 4.2.1, while the second one is studied in Section 4.2.2.

4.2.1 Pseudonym Certificate Distribution

The basic mechanisms of standardized PSC distribution have been introduced in Section 2.2.4.3. However, prior work has not studied the individual influences of the dedicated sub-mechanisms of the overall PSC dissemination algorithm. Moreover, several details of the standardized algorithm show ambiguities, as outlined in the following. Topics covered in this section are partly covered by prior work of the author in [32][3].

[3]Contribution of co-authors mainly relates to implementation of considered traffic scenarios and parts of the considered set of PSC distribution mechanisms within the simulation environment. The main contribution is from the author of this work.

In general, the standardized PSC distribution algorithm can be separated into three distinct sub-mechanisms, which are

1. neighborhood aware PSC emission, which can be regarded as an implicit request scheme, i.e., each message serves as an implicit request to respond with including a PSC in case the sender is found to be a new neighbor. Two variants of this mechanism can be used, which are

   (a) an unsecured variant, which detects a new neighbor also based on unverified messages not including a PSC, and

   (b) a secured variant detecting a new neighbor only based on a message, which could be verified, i.e., it included all certificates required for its validation.

   ETSI ITS and WAVE standards use variant no. 1a [125, 176].

2. Explicit requests are used via an optional and variable length list of requested certificates. This list is included on-demand in the security envelope. The maximum number of entries is limited to six in both ETSI ITS and WAVE [125, 176]. Requests for both PSCs and AACs may be present. However, there is hardly any comment on how to manage this list within the standards and prior work has also not looked at this aspect in detail. The main question is how to remove entries from the list. This problem can be separated into two main aspects as follows.

   (a) How to handle situations in which more unknown PSCs should be requested than there are free entries in the list? One can either

      i. drop new entries in case the list is full,

      ii. buffer new entries in a second, longer list and keep existing entries, or

      iii. maintain the list in a first in first out (FIFO) manner, i.e., a new entry replaces the oldest one.

   (b) How often should a request be sent? Including the choices of

      i. including a request only once (remove after sending), or

      ii. repeating a request. Multiple possibilities exist for this strategy including

         A. repeating until the request gets answered, or

         B. repeating for a fixed time (remove by timeout), or

         C. repeating for a fixed number of requests, or

         D. a combination of no. 2(b)iiB and 2(b)iiC.

      Strategy no. 2(b)iiA is not recommended in VANETs, because of typically high node mobility and limited communication ranges. This leads to situations in which only a single packet is exchanged between two nodes. Such situations can lead to an unlimited repetition of explicit requests, which causes pure overhead.

3. Cyclic PSC distribution includes a PSC after a timeout elapsed, e.g., after the PSC has not been included during the last second by some other mechanism [125].

To evaluate the different possibilities to create a standard conforming PSC distribution strategy, a freeway and an urban roundabout scenario are considered. For details about these scenarios see Section 3.2. In both scenarios all of the following PSC dissemination strategies are considered. These include,

1. ETSI ITS based PSC emission (see also Section 2.2.4.3) with repeated explicit requests,
2. no. 1 without repeated explicit requests, i.e., onetime requests,
3. no. 1 without unsecured implicit requests,
4. no. 2 without unsecured implicit requests,
5. no. 1 without any implicit requests,
6. no. 2 without any implicit requests,
7. no. 3 without any explicit requests, i.e., only secured implicit requests.

The above given numbering scheme is used to refer to the individual strategies in the remainder of this section.

Approach no. 7 is the only sporadic PSC emission scheme, which does not require to make use of any data set from a message whose digital signature cannot be checked, due to unavailability of the sender's PSC. Thus, an attacker without access to valid ITS credentials cannot influence this PSC dissemination scheme. In contrast, this is possible for proposals no. 1 to 6. The DOS style attack on PSC dissemination from Section 5.1.1 misuses this property.

The core aim of implicit and explicit PSC request schemes is to minimize cryptographic packet loss by enabling fast mutual authentication between new neighbors. Hence, an evaluation considering the extend of cryptographic packet loss in two different traffic scenarios under presence of PSC request schemes no. 1 to 7 is provided in the following.

Evaluation results for the freeway scenario (see also Section 3.2) are illustrated in Figure 4.3. The displayed values clearly show that approach no. 1 outperforms its counterparts, as it yields the lowest message discarding rate, i.e., cryptographic packet loss, for all traffic densities. The provided error bars in Figure 4.3 represent the standard deviation, as calculated from collected measurement values. Lower values of the node interval lead to increased traffic density. Hence, the traffic density increases in Figure 4.3 from the left to the right.

One can clearly see an increase in the message discarding ratio caused by disabling the unsecured implicit PSC request mechanism (no. 3 vs. no. 1 and no. 4 vs. no. 2). Completely disabling implicit requests further decreases system performance, as exhibited by results for no. 5 vs. no. 3 and no. 6 vs. no. 4. The only scheme which is able to completely avoid cryptographic packet loss by design is an always include strategy. However, the channel load caused by that strategy has been found to be much to high in prior work, see e.g., [133, 135]. Schemes no. 1 to 7 are found to not show a statistically significant difference in regard to caused channel load. Hence, no details about this metric are given here.

Measurement results for the roundabout scenario (see also Section 3.2) are given in Figure 4.4. In general, results for the roundabout scenario are pretty similar to the ones of the

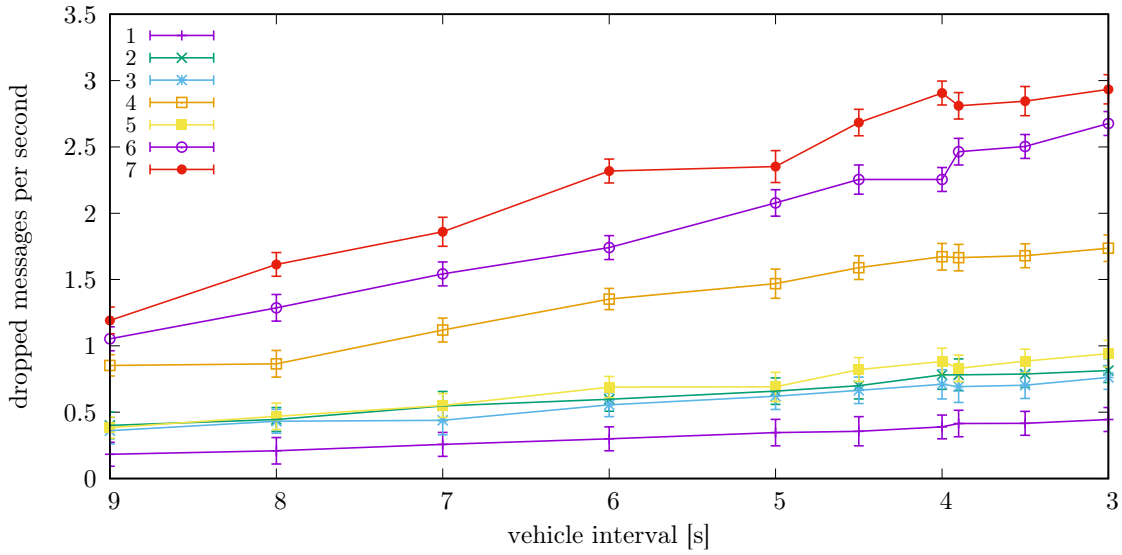


Figure 4.3: Cryptographic packet loss in the freeway scenario.

freeway scenario discussed before. However, one can see from the comparison of both scenarios that the much lower traffic density also leads to a significant decrease in the number of discarded messages per second.

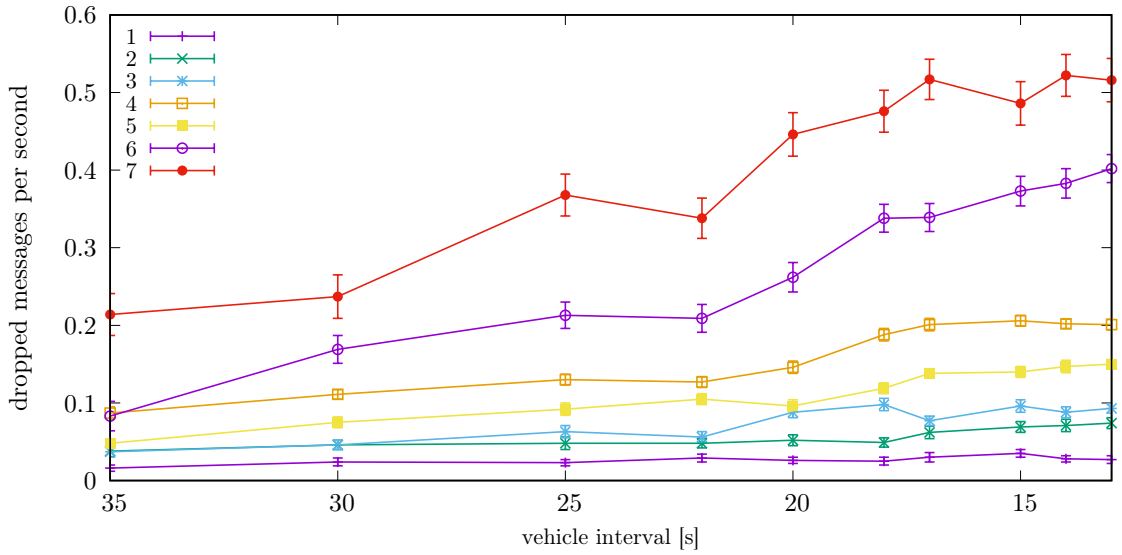


Figure 4.4: Cryptographic packet loss in a roundabout scenario.

All obtained results show that strategy no. 7 leads to the worst system performance in comparison to the other considered approaches. Thus, found results show a significant dependence of effective PSC dissemination on using unverified information, i.e., unsecured requests. Hence, disabling this mechanisms to guard the system from attacks leads to significant system perfor-

mance degradation in ordinary (i.e., non attacked) use cases.

Moreover, the evaluation results show that strategy no. 1 leads to the lowest amount of found cryptographic packet loss. Furthermore, small standard deviations of measured values illustrate well dependability of the mechanism. Hence, usage of a combination of unsecured implicit and repeated explicit PSC requests is recommend for ETSI ITS.

4.2.2 Certificate Chain Distribution

Since version 1.2.1 of the ETSI ITS standard for the security envelope, on-demand exchange of certificate chains between nodes is supported [109,125]. WAVE uses a similar mechanism [176]. The basic mechanisms of AAC requests are introduced in Section 2.2.4.3. Prior work has not studied the efficiency of the standardized AAC dissemination mechanism in detail. Thus, such an evaluation is provided in the following. The topic is also covered by prior work of the author in [39,41][4].

The used traffic scenario for all simulations is the freeway scenario, which is described in-detail in Section 3.2. Unfortunately, there are no reference scenarios for AAC or certificate chain distribution in prior work. Moreover, the used core zone concept filters out the initialization (or edge) effects created by injecting new nodes into the simulation in the freeway scenario. Thus, we use the following concept to evaluate AAC requests during simulation runs.

Four different cases of AAC requests are considered. These consist of the scenarios that a requester requests

1. one AAC and all receivers answer the request (worst case),
2. the maximum of six different AACs being answered on average by 61.06 % of all receivers,
3. the single most common AAC, thus the request is answered on average by 21.31 % of all receivers, and
4. one randomly picked AAC causing a request, which is answered on average by 8.60 % of all receivers.

To evaluate cases no. 2 to 4, AACs are distributed according to Original Equipment Manufacturer (OEM) sales figures for Germany given in [198]. In doing so, a common AAC for each OEM's vehicles is assumed. This leads to the given percentages of nodes, which can be expected to respond to the different kinds of requests in cases no. 2 to 4. In contrast, different OEMs are assumed to always use different AACs. In contrast, a common AAC is used for all nodes to show the impact of case no. 1.

An AAC request increases the average message size of CAMs within a requester's surrounding by causing an increase of the security envelope's size by a factor of $i_{AACreq.}$, over the ordinary CAM security envelope's size (without a received request). Regarding only cyclic PSC

[4]Contribution of the co-author is mainly in regard to the methodology of implementing the considered mechanisms within the simulation environment. The main contribution is from the author of this work.

inclusion into CAMs, an upper bound on the increase can be determined by

$$i_{AACreq.} \leq \frac{s_{\text{CAM,PSC+AAC}} \cdot p + (1-p) \cdot \overline{s}_{\text{CAM}}}{\overline{s}_{\text{CAM}}}. \tag{4.2}$$

The size of a CAM's security envelope holding a certificate chain is given by $s_{\text{CAM,PSC+AAC}}$ (= 349 bytes), and the one of an average CAM's security envelope by $\overline{s}_{\text{CAM}}$ [125]. The share of nodes reacting to the request is given by $p \in [0; 1]$.

An upper bound on $i_{AACreq.}$ can be obtained as follows. $\overline{s}_{\text{CAM}} = 105.5$ bytes holds for 10 Hz CAM emission frequency and minimal 1 Hz PSC inclusion frequency [119, 125]. Thus, $\max(i_{AACreq.}) = 3.32$ is an upper bound on the increase in average message size for ETSI ITS ($p = 1$). The bound is matched in case of no present implicit or explicit PSC requests in the VANET. Otherwise, PSC inclusion happens more frequently. Such extra PSC emissions increase $\overline{s}_{\text{CAM}}$. Hence, $i_{AACreq.}$ is caused to be smaller than the given bound.

The amount of PSC requests depends on the traffic scenario, as such requests happen when the surrounding of a node changes. Hence, the experienced value of $i_{AACreq.}$ depends on the traffic scenario, too.

Table 4.7 gives theoretical bounds as well as simulation results for the CAM size increase at the position of the requester during a time span of 100 ms after a single request message got sent. The requester is inserted as an RSU into the simulation at its center using the freeway scenario from Section 3.2, after the ordinary traffic flow was built up. An interval of five seconds between successive requests is used to ensure that the individual requests do not influence each other. An average node interval of three seconds per lane is used to obtain the measured results in Table 4.7. This leads to an average PSC inclusion interval of about 3.0 Hz and an average request list size of 2.3 entries. Each entry in the request list is 3 bytes long, and the presence as well as the length of the list is both encoded with a one byte long data field. Hence, $\overline{s}_{\text{CAM}} = 139.4$ bytes holds for the measured results.

	worst case	6 most common AACs	most common AAC	average AAC
bound	3.32	2.42	1.49	1.20
measured	2.51	1.92	1.32	1.13

Table 4.7: Security envelope size increase factor after an AAC request.

Results given in Table 4.7 show that even requests for a single AAC causes the average message size to increase significantly. Moreover, a node having no prior knowledge about other nodes AACs will request multiple AACs with high probability. Thus, the found increase will be significant, and happen multiple times, as more than just six AACs can be expected to be used in practice.

The following section discusses the problem of cross-layer size restrictions and their impact on the message assembling procedure inside an ETSI ITS protocol stack.

4.3 Cross-layer Size Restrictions

Within current VANETs' protocol stacks, assembling of the data sets used by the different protocol layers for beacons (i.e., CAMs or BSMs) is independent. This means, the individual protocol layers do not coordinate the inclusion of optional data sets. Additionally, stateless VANET communication for safety critical use cases does not use message fragmentation [122, 174]. This combination of features can lead to problems for message transmission on the MAC layer, due to rigid maximum message size restrictions used to limit the channel load in VANETs. ETSI ITS uses a limit of 650 bytes, which is part of the always active part of the DCC mechanism [103]. Topics covered within this work is partly covered by prior work of the author given in [44][5].

A common approach from other network protocol stacks is to use significant data size variations only on the application layer, e.g., IP-based protocol stacks [69]. Moreover, the higher protocol layers always leave enough spare message size for lower layers to allow them to include their maximum size data sets. However, this is not the case for current VANETs, as shown in the following.

Sizes of data sets on the different ETSI ITS layers with standardized data encoding rules are given in Table 4.8. The shown data size requirements for BTP, GeoNetworking service primitives Single Hop Broadcast (SHB) (used for CAMs), Geo-Broadcast (GBC) (used for DENMs), and the access layer (Logical Link Control (LLC) and MAC) are constant, at least for a dedicated message type. More GeoNetworking service primitives have been defined, but standard use cases do not make use of them so far [122]. Hence, they are not given in Table 4.8.

layer	protocol	minimum size	common size	maximum size
7	CAM	42	389	428
	DENM	41	397 / 607	2681
4	BTP	4	4	4
3	SHB	40	40	40
	GBC	56	56	56
	sec. CAM	93	218(+20)(with PSC)	340
			350(+20)(with PSC + AAC)	570
	sec. DENM	229	229	331
	sec. generic	229	229	331
2	LLC	8	8	8
	MAC	30	30	30
sum CAM		217	**689(+20)**	**850**
			821(+20)	**1080**
sum DENM		368	**724 / 934**	**3110**

Table 4.8: Data field sizes of protocol layers within ETSI ITS and ITS-G5.

To obtain the given numbers in Table 4.8 standards [103, 119, 120, 122, 125] have been used. Figures in the first column slightly differ to the ones from Table 4.2, as those results refer to

[5] See also footnote 1.

[109], i.e., version 1.1.1 of the standard and not version 1.2.1, which is used for Table 4.8.

The results from Table 4.8 clearly show that many messages combined from data sets, which on their own semantically and syntactically comply to their standards, lead to very high message lengths. Thus, there is a conflict with the maximum message size of 650 bytes defined within the ITS-G5 DCC rules (see also Section 2.1.2). Hence, such messages are dropped by any ITS-G5 conforming MAC layer implementation. In Table 4.8 affected messages are marked in boldface.

In case of WAVE, the problem described above for ETSI ITS does not affect transmission of BSMs. The 802.11p MAC layer packet size limit is not as strict as the one of ITS-G5 [169]. However, the basic mechanisms of message assembly are the same for WAVE and ETSI ITS. Hence, sending of larger messages within a WAVE protocol stack suffers from the same problems, as the ones outlined for ETSI ITS above.

Data size requirements from the security envelope as well as from the facility layer are discussed in detail in the following two sections.

4.3.1 Data Size Requirements Inside the Security Envelope

For the security envelope a large range of data lengths is possible, as shown in Table 4.8. Thereby, the range starts from 93 bytes (no PSC) and goes up to 350 bytes (with PSC and AAC, i.e., a certificate chain) for CAMs, even in case only the mandatory data fields are present. Each one of the given values can be extended by up to 20 bytes, in case the optional so called *certificate request list* with up to six entries and 3 bytes per entry is present. This list is not used for other message types [125].

For DENMs, the size of the security envelope is always 229 bytes at least. For all remaining message types (security profile generic) 229 bytes are used at least, too. Both security profiles specify to always include the PSC, but never the AAC [125]. Since version 1.2.1 [125], there is no more difference in regard to the included data sets between security profiles DENM and generic. To obtain minimal and common values the mandatory header fields for the security envelope and the certificate have been used.

Moreover, the ETSI ITS standard (and also its WAVE counterpart [176]) for the security envelope format allows to include many more optional data fields in a certificate, e.g., extra validity restrictions [125]. These data sets have been used to obtain the worst case sizes in Table 4.8. Thereby, two ITS-Service Specific Permissions (SSPs) (for CAM and DENM, $2 \cdot 4$ bytes) have been used. Additionally, a location restriction given by a polygonal region (98 bytes) is included. Other location restrictions were found to yield a significantly lower size of the security envelope.

The obtained results clearly show that the size of the security envelope is highly variable. This can lead to an excess in overall message size. Consequences of this finding are discussed in Section 4.3.3, after the discussion of the data size requirements within the facility layer in the next section.

4.3.2 Data Size Requirements Inside the Facility Layer

For CAMs, the data sizes from Table 4.8 are obtained as follows. A message with 42 bytes holds only mandatory data, 389 bytes contain mandatory data and a low frequency container, and up

to 428 bytes are used when all data fields are set in high frequency, low frequency and public transport container. Please note that the public transport container is only one possible choice for the special vehicle container. However, we found it to be the most data size consuming one. The standardized CAM assembly rules specify to include only one optional container, i.e., only low frequency or public transport container, but this is not enforced by the ASN.1 specification of CAMs [119].

The biggest share of data length within a CAM is required by the so called path history field within the low frequency container. It holds 40 2-D positions (GNSS coordinates) with corresponding confidence values. Current standards do not specify how to fill this data field, but the C2C-CC's basic system profile calls for covering the last 200 m to 500 m of traveled trajectory within this data field [60]. The size of a common size CAM is such high that a resulting message is always dropped at the MAC layer of ITS-G5. This is discussed in more detail in Section 4.3.3.

The size of DENMs is even higher in comparison to CAMs. The only exception is the case of a minimum size DENM. However, usage of this kind of DENM does not make sense in practice, as it only holds the management container and no usable information about the event itself.

Two results are given for a common size DENM in Table 4.8. The smaller one is obtained from a DENM containing a situation and a location container, which are always included together [120]. Thereby, within the situation container only mandatory fields are present, e.g., no event history field is included. Moreover, the location container includes only one path history field. Up to seven path history fields may be contained at once [117].

The second value for a common size DENM is the data size obtained in case the optional values inside the situation container are used. Summing up the obtained DENM length with minimum overhead of lower layers shows that the obtained message always exceeds the standardized packet size limit at the MAC layer. Hence, it will drop all such DENMs, although these use a syntactically correct and semantically reasonable combination of data fields at the facility layer.

For the worst size DENM, all possible data fields have been used. One can see from Table 4.8 that this leads to an excess in message size. MAC and network layer maximum payload limits are greatly exceeded. Hence, such a message is dropped by these layers.

The obtained results show that inclusion of long position sequences in CAMs and DENMs causes massive conflicts between required message size and payload size restrictions. Thus, significant limitations of the length of these data fields are required to allow a standard conforming dissemination of CAMs and DENMs.

4.3.3 Cross-layer Data Size Conflicts

The obtained worst case size of the security envelope for CAMs shows a design weaknesses of the protocol stack. With a 570 bytes long security envelope, it is not even possible to send a BTP packet without any further payload. The sum of all protocol layers' overheads already exceeds the maximum packet size of the MAC layer. A main contributor is the polygonal region validity restriction of certificates, which holds 12 GNSS positions, with four bytes each (see also

Section 4.3.1). Hence, this kind of validity restriction should be removed, as its usage renders the remaining system useless, unless the accepted maximum packet size is significantly increased.

Layer two to four overhead for CAMs commonly sums up to either 175 bytes without included PSC, 299 bytes in case of an included PSC and even 431 bytes in case of a contained certificate chain (see Table 4.8). However, in case of a CAM with present low frequency container, the facility layer leaves only 261 bytes to lower layers. This only works out in case the security entity does not include any certificate. Currently, certificate inclusion is not done in a data length aware manner [125]. Therefore, violation of the message length limit, which leads to discarding of messages at the MAC layer, can occur.

In case of a low CAM generation rate of $f_{CAM} = 1$ Hz, all the mentioned optional data fields (CAM's low frequency container and PSC) are always present in each message [119, 125]. The timeout interval to include the low frequency container is 500 ms [119]. This can lead to a situation in which the affected node cannot transmit any CAM, as these are all dropped at its access layer. Hence, this has to be regarded as a severe design weakness of the current ETSI ITS protocol stack. The CAM discarding rate d_{CAM} for $f_{CAM} > 1$ can be calculate by

$$d_{CAM} = \frac{f_{cert}}{f_{CAM}} + \frac{f_{cert}}{f_{CAM}} \cdot \frac{f_{cert} - 1\text{ Hz}}{f_{CAM} - 1\text{ Hz}} + \left(1 - \frac{f_{cert}}{f_{CAM}}\right) \cdot \frac{f_{cert}}{f_{CAM} - 1\text{ Hz}} \cdot 1\text{ Hz}. \quad (4.3)$$

f_{cert} gives the PSC inclusion frequency. Please note that due to the used PSC piggybacking strategy $f_{CAM} \geq f_{cert}$ holds. Furthermore, it is assumed that inclusion of low frequency container is statistically independent from inclusion of the PSC. This is reasonable, as within ETSI ITS there is no coupling between the inclusion rules for both data sets.

The minimum inclusion frequency of PSCs $\min(f_{cert}) = 1$ Hz leads to $\min(d_{CAM})$. Corresponding values are given in Table 4.9. For an exemplary PSC emission rate of $f_{cert} = 3$ Hz (obtained from the freeway scenario described in Section 3.2), values for d_{CAM} are also given in Table 4.9.

f_{CAM} in Hz	10	9	8	7	6	5	4	3	2	1
$\min(d_{CAM})$ in Hz	$\frac{1}{5}$	$\frac{2}{9}$	$\frac{1}{4}$	$\frac{2}{7}$	$\frac{1}{3}$	$\frac{2}{5}$	$\frac{1}{2}$	$\frac{2}{3}$	1	1
$\min(d_{CAM})$ in %	2	2.5	3.1	4.1	5.6	8	12.5	22.2	50	100
d_{CAM} in Hz ($f_{cert} = 3$ Hz)	$\frac{3}{5}$	$\frac{2}{3}$	$\frac{3}{4}$	$\frac{6}{7}$	1	1.2	1.5	2	2	1
d_{CAM} in % ($f_{cert} = 3$ Hz)	6	7.4	9.4	12.2	16.7	24	37.5	66.7	100	100

Table 4.9: Average share of CAMs discarded due to a maximum packet size violations at the access layer.

Under presence of an attacker who wants to perform a DOS attack the situation is even worse. Rapid certificate (chain) inclusion in (almost) every CAM can be caused, as explained in detail in Section 5.1.1 and 5.1.2. Hence, a node can be banned from sending any CAM in case it is attacked and subject to the above described internal message discarding. This clearly makes the attacker achieve his goal of performing a DOS attack.

Protocol stack behavior in both cases, with and without presence of an attack, clearly shows that the message assembling approach, which has been used so far, needs to be improved. Hence, an alternative is suggested in Section 4.3.4.

Moreover, the found cross layer data size issues questions the usability of a PSC omission approach, like the one suggested in [133, 135]. Such kind of PSC dissemination strategy tries to achieve $f_{cert} = f_{CAM}$ and only omits PSC emissions in case the channel load is found to be too high. However, this leads to a massive amount of messages getting dropped due to maximum size violations at the MAC layer.

Basically, three different possibilities exist to overcome the found message length problem without changing the sporadically included content itself. Theses are

1. to increase the maximum allowed message size at the access layer by either
   (a) increasing the maximum air time T_{air} (see [103]) of a single packet, or
   (b) increasing the fixed transmission data rate of the control channel at the physical layer, or
2. to use packet fragmentation, e.g., at the network layer,
3. to coordinate inclusion of sporadically included large data sets between different protocol layers to efficiently share message content resources.

Approaches 1a and 1b would significantly change the characteristics of the VANET's wireless channel. With increased T_{air} both the probability of packet collisions and the channel load created by a transmitted packet increase. Moreover, increasing the transmit data rate on the physical layer makes the system less robust against common challenges of wireless communication, e.g., signal distortion. Hence, both approaches can be expected to significantly reduce the average communication range of nodes. Therefore, such kinds of approaches are not recommend.

Packet fragmentation support for VANETs is studied in [62] assuming package reception is acknowledged by receivers. However, this is not the case in broadcast mode of ETSI ITS and WAVE approaches. [158] studies the trade off between large packets with no or low amount of fragmentation and many short packets, as a result of massive fragmentation. Thereby, it is found that optimal packet length depends on traffic conditions, and should be smaller than 1000 bytes for typical traffic densities. Hence, massively increasing T_{air} seems infeasible to overcome the found packet size issue. Moreover, the influence of fragmentation on delays for data distribution, which influences the cooperative awareness quality of nodes, has not been looked at. However, increasing the number of packet transmissions by fragmentation will increase the channel load on the highly bandwidth restricted single control channel, due to significant protocol overhead contained in each packet. One can assume that the problems encountered in case of fragmentation support show similarities to the ones shown in the case of IP-based communication [187]. Therefore, inclusion of fragmentation support into VANET approaches seems unlikely in the near future.

Instead, a cross-layer content aware message assembling strategy, as suggested in no. 3, is recommended. Hence, such a strategy is introduced in Section 4.3.4.

4.3.4 Cross-layer Size Aware Packet Assembly

To enable message assembly, which takes into account cross-layer data size requirements, knowledge about message size limitations and the minimum data size requirement from each

layer has to be available to all higher layers. Thus, the management entity (see Figure 2.2) should collect the requirements from the individual layers and disseminate it to all of them. Thereby, the available message size is to be obtained from the access layer chosen by the network layer. In a hybrid communication scenario, the dissemination technology can be chosen for each message individually. Thus, the available maximum message size may also vary between packets.

Awareness about the individual layers' needs for inclusion of variable length data sets throughout the protocol stack is required to overcome the found problems. Hence, each layer has to provide information about the minimum size of data it has to include in a dedicated message ($m_{layer}^{message\ type}$). However, this limit may vary between packets. To keep the individual layers clearly separated, this information should be available through the cross-layer management entity. For example, it should provide the facility layer's CABS with information about the minimum data size requirement from lower layers for the next CAM to be sent. The facility layer only needs to know about this summed up value. Hence, the individual composition of the reserved data size can and should be abstracted by the management entity.

Within ETSI ITS and WAVE the minimum data size required by a protocol layer only depends on the message type, e.g., GeoNetworking uses seven different but fixed formats/sizes of the extended header [122]. Fortunately, no cross-layer dependencies between presence of optional fields on the different layers exists. For example, the PSC gets included independently from the presence of optional containers in a CAM. Hence, once generation of a new message has been triggered, which typically happens at the facility layer, the management entity can gain knowledge about lower layers' message part consumption requirements just based on the message type. One should note that this may not hold for other protocol stacks, which would significantly increase the effort of determining lower layers' requirements.

Two different approaches to the content and length aware message assembling problem are discussed in the following. The first one is a simple top down approach (Section 4.3.4.1), while the second proposal uses bottom up reservation of packets' parts (Section 4.3.4.2).

4.3.4.1 Strict Top Down Approach

A straight forward approach is to just have each protocol layer take its share of a packet's maximum size without taking variable length of lower layers into account. Thereby, each layer has to make sure that the remaining unused share of the packet is at least long enough to hold the minimum length data fields of lower protocol layers. Thus, lower level entities have to cope with the unused packet's part left over by higher level entities.

However, this approach faces a major drawback. It can lead to starvation of lower protocol layers. In case higher layers always leave just the minimum required message share to lower layers, these lower layers can never include their required, sporadically included data sets. Thus, distribution of extra (meta) data is not possible, even in case it would be required to support further communication.

Regarding ETSI ITS, this approach leads to the following scenario. A CAM generation frequency of 1 or 2 Hz always leads to inclusion of the optional low frequency container in every CAM [120]. Hence, the security entity will never be able to include a certificate (chain), and distribution of PSCs and AACs will not work at all. Hence, all messages will be dropped

by their receivers, due to cryptographic packet loss. Thus, receivers will never achieve awareness of affected other nodes on their facility layer. However, the affected node will be able to send out messages, in contrast to the case without maximum length awareness being currently standardized.

Due to the discovered drawback of the simple top down mechanism, another approach using message part reservation from lower layers at higher layers is discussed in the following.

4.3.4.2 Top Down with Bottom Up Reservation

Basically, the approach proposed in the following works like the one from Section 4.3.4.1. However, we add the possibility for lower protocol layers to reserve space within messages at higher layers.

As lower layers, e.g., the network layer, should not be required to know about the existence of specific higher layers, they should not reserve message parts directly at such higher layers. Instead, we propose a publisher and subscriber mechanism run by the management entity as follows. Each layer, except of the highest one, which needs more than its minimum amount of message share, informs the management entity about its required data length. The management entity will then inform all subscribers about this kind of request. All layers, which enlarge the size of a message, have to subscribe at the management entity to be informed about such announcements.

After having received a data length reservation announcement, the higher level entities should make sure that they leave the requested spare data size to lower layers. One could try to introduce some kind of prioritization into that procedure. Thereby, the higher layers could ignore the reservation requests in some cases. However, this introduces dependencies between different layers which will be hard to maintain, especially in case of hybrid communication scenarios with a common higher layer and multiple lower layers, e.g., at the access layer level. Hence, such priorization attempts are not considered in the following.

Message Part Allocation Algorithm For the design of the reservation algorithm, one has to take into regard when functionalities at different layers know which amount of data they are about to sent in the next message. Thereby, one can differentiate two characteristics, which are

1. on the fly decision, i.e., decision is only made when a new message gets assembled, or
2. asynchronous decision, i.e., decision can be done in advance, e.g., triggered by received messages.

In case only strategy no. 1 is used, the system will be equal to the one in Section 4.3.4.1. Higher level entities just use as much as they want to do and lower levels have to cope with the remaining spared message part.

Therefore, strategy no. 2 should be used whenever possible. This means, that whenever a criteria for including optional data gets fulfilled, e.g., a timeout for including cyclically distributed data happens, the corresponding required message part gets reserved at higher layers.

One can assume that an entity, which is not able to transmit its optional data in the current message, will try to do so again in the next one. Additionally, a communication connection is

typically built in a bottom up nature, i.e., the link has to be maintained at the network layer level to allow message exchange at higher layers. Thus, a failed message part allocation should be repeated straight after the message for which the failure to include occurred was sent at the lowest layer. Thereby, the lowest layer is allowed to perform its reservation first. Thus, the probability of successful message part allocation will be high, as other entities will have this possibility later.

To avoid that the asynchronous reservation mechanism blocks the bottom up reservation mechanism from working, such requests should only be allowed in case no message sending is currently due at layers below the requester.

The proposed message assembling procedure is illustrated in Figure 4.5 for the case of a CAM. At t_{CAM} the facility layer's CABS decides to send a CAM (trig. CAM). It informs the management entity about its decision to obtain the maximum size of contents, which it can sent in this CAM. The management entity reacts with requesting the required content sizes of all lower layers (req. min CAM size). Each layer determines its content size requirement within the next CAM and reports the result to the management entity ($m^{CAM}_{\{tra,net,sec,acc\}}$). Then, the management entity informs each layer (above the access layer) about the cumulated data size requirements of the layers below it. Finally, the CAM is generated and each layer adds its content to it, before it is sent.

After a CAM from an unknown node, i.e., from a new neighbor, is received at t_{new} (CAM new node), the security entity decides to send its PSC in the security envelope of the next CAM. Thus, it requests to allocate the corresponding size within the next CAM at the management entity (req. m^{CAM}_{sec}), as shown in the lower part of Figure 4.5.

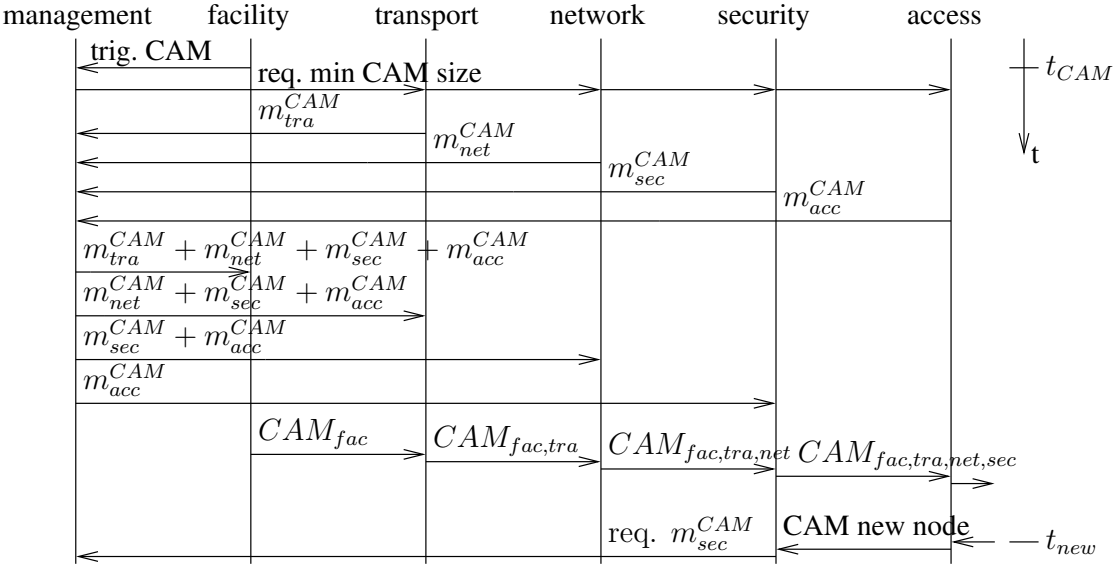


Figure 4.5: Cross-layer sequence for CAM assembling.

A typical inclusion sequence of both the low frequency container inside CAMs (facility layer) and the PSC within corresponding security envelopes (security entity) in case of used

cross-layer coordination is given in Figure 4.6. Thereby, the intention to include optional data, i.e., either the low frequency container or the PSC, in the next message is given by a dashed line. Massive dots show the points in time at which a new CAM gets assembled.

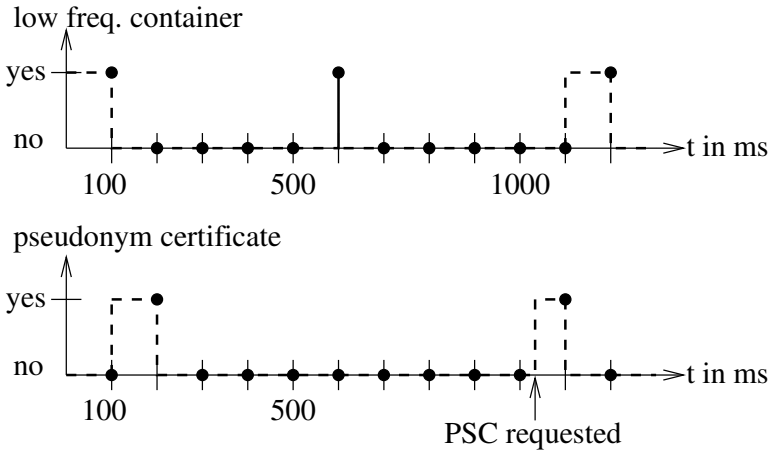


Figure 4.6: Cooperative inclusion of sporadically distributed data sets within CAMs at the facility layer (upper part) and security envelopes (lower part).

In the example from Figure 4.6, a node starts up at $t = 0$ ms. The CABS on the facility layer and the security entity both want to include their optional data in the first CAM. The facility layer can do so first ($t = 100$ ms), while the security entity has to wait for the second generated CAM ($t = 200$ ms). At $t = 600$ ms, the CAM contains optional data from the facility layer again. A message part reservation by the security entity happens at $t = 1030$ ms, which delays the inclusion of optional data at the facility layer for one CAM. Thus, the PSC is sent at $t = 1100$ ms, while the low frequency container gets disseminated at $t = 1200$ ms.

Starvation of High Protocol Layers Unfortunately, with the approach from the section before, starvation of high protocol layers is possible. For example, the CAM's low frequency container can only be included with $f_{low.freq} = 2$ Hz in case $f_{CAM} - f_{cert} \geq f_{low.freq}$ holds, i.e., in case there are enough CAMs without an included PSC, hence being able to hold a low frequency container. Moreover, in case of a present PSC or AAC dissemination attack (see also Sections 5.1.1 and 5.1.2) CAMs can never include optional containers. Hence, all the information from those containers will not be disseminated and applications relying on them will not be able to work. However, the bottom up reservation approach ensures that all protocol layers up to a certain level (for ETSI ITS the network layer security entity) will work well. In contrast, this is not ensured by the proposal from Section 4.3.4.1.

One should note that lower protocol layers can typically work without support from higher level ones, but this does not hold vice versa. Hence, the approach from this section should be preferred in comparison to the one from Section 4.3.4.1.

As mentioned above, higher layers could reject message part allocations from lower one to enforce inclusion of their own data sets. However, we propose to limit data sizes requirements at lower layers instead.

The following section studies mutual influence of certificate distribution and pseudonym change strategies.

4.4 Cross Influence between Certificate Distribution and Pseudonym Change

Certificate distribution and pseudonym change are two subjects within the VANET domain, which have been studied extensively in prior work. However, there is a lack of work taking into regard the mutual influence of both mechanisms. Hence, an in-detail study about such kind of cross influence is provided in the following. Topics covered in the following are also part of prior work of the author in [42][6].

Mutual influence of PSC dissemination and pseudonym change, i.e., PSC change, can be initiated from both sides. Thereby, analysis of current standards [125, 176] shows that

1. pseudonym change always
   (a) causes the need to distribute the new PSC of the node, which changed its pseudonym, and
   (b) triggers new node detection at other nodes receiving the new PSC. According to the standardized certificate distribution algorithms, this causes inclusion of the PSC into the very next beacon message of every receiver [125, 176]. Thus, certificate changes may undermine the bandwidth saving approaches of developed sporadic certificate distribution strategies.
2. PSC distribution can cause the detection of an address duplication. [54] suggests to change the used PSC in case the lower 32 bits of a node's own certificate ID (i.e., its station ID) are identical to the ones of a received certificate's ID (see also Figure 2.4). The certificate change then again causes influence no. 1.

One should note that PSC distribution strategies not using neighborhood aware PSC emission are not affected by case no. 1b. One example of such kind of strategy is channel load dependent PSC omission [133, 135]. However, such strategies have not been considered for usage in ETSI ITS or WAVE.

The probability of case no. 2 is quite small, due to the quite large size of the station ID. In contrast, case no. 1 happens regularly during standard operation of a VANET. Moreover, case no. 2 always causes case no. 1. Thus, our focus is on case no. 1 in the following.

4.4.1 Uncoordinated Pseudonym Change

In case of uncoordinated pseudonym changes, an equal distribution of such changes over time can be assumed. A node changes its PSC after a timeout, which is defined by a fixed time span varied by a small random value to avoid synchronization effects enabling node tracking

[6]Contribution of the co-author mainly relates to providing support during implementation of the considered schemes within the simulation environment. The main contribution is from the author of this work.

[104, 154, 205, 251]. Without coordination between nodes, there will be steady presence of PSC changes going on in the VANET. Thereby, the amount of such changes experienced by a single node depends on the amount of other nodes within its communication range and the PSC change interval.

The number of PSC changes experienced by a single node within a time interval i between two successive beacons is denoted by c_i. Assuming a fixed beacon frequency f given in Hz (in WAVE fixed to $f = 10$ Hz),

$$c_i = \frac{n_{i,known}}{\bar{t} \cdot f} \tag{4.4}$$

holds. t denotes the time interval between PSC changes in seconds, and $\bar{t}$ gives the average of t. As it does not make sense to change the pseudonym faster than sending of messages is performed, $t > \frac{1}{f}$ holds. It is assumed that pseudonym changing times are not synchronized between nodes, i.e., pseudonym changes are evenly distributed over time. n_i gives the number of nodes within communication range of a node, i.e, those from whom a message is received within interval i. These nodes consist of two disjoint sets $N_{i,known}$ and $N_{i,new}$, which represent the nodes whose identities have been known in the past and those who have not been known to the receiver, respectively. For them

$$n_i = n_{i,known} + n_{i,new}; n_{i,known} = |N_{i,known}|, n_{i,new} = |N_{i,new}| \tag{4.5}$$

holds. The receiver only experiences the pseudonym change of another node in case it knew about this node before the change happened, i.e., in case it belongs to $N_{i,known}$. In contrast, nodes from $N_{i,new}$ are always new neighbors for the receiver, independently from whether they just changed their pseudonym or not.

In practical scenarios, c_i is not a fixed value, but changes frequently, as both the communication range of a node as well as surrounding traffic density vary. Moreover, within ETSI ITS f can vary between 1 Hz and 10 Hz between nodes and over time.

Figure 4.7 gives an illustration of values for c_i, which can be expected in practice. Thereby, a node density from 1 up to 400 nodes within communication range of a single node is assumed [279]. Certificate change intervals of 10 s, 30 s and 5 minutes (= 300 s) are considered following recommendations in [154, 276, 325]. A fixed beacon frequency of $f = 10$ Hz is assumed, as it is used in WAVE.

Provided values in Figure 4.7 show that c_i significantly increases with shorter pseudonym change intervals. Thus, the impact of pseudonym change on neighborhood aware PSC distribution is clearly lower for higher values of t. However, high values of t result in lower levels of privacy for nodes [276, 325].

Results in Figure 4.7 show that for high node densities, $c_i \geq 1$ holds. Due to neighborhood aware PSC distribution, this means that one can expect inclusion of the PSC in (almost) every emitted beacon message. Multiple newly detected neighbors between sending of two beacons do not further increase the emission frequency in comparison to a single new neighbor (saturation effect). Thus, a PSC inclusion frequency, which is almost equal to f, can be expected.

Results from prior work show that much smaller PSC inclusion frequencies have to be used to achieve required cooperative awareness among nodes [33, 133, 135]. This holds especially for high density traffic scenarios. The found extra inclusions of PSCs undermine the bandwidth

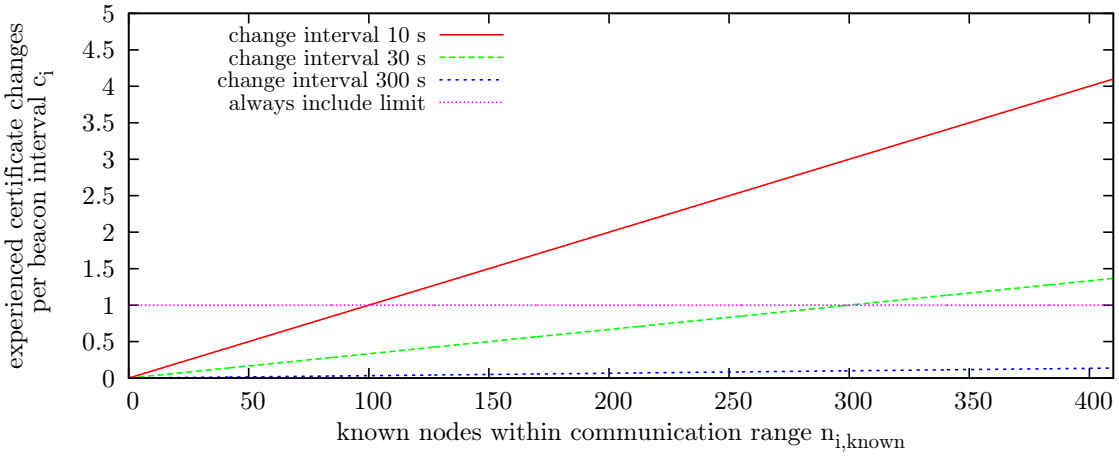


Figure 4.7: Number of new nodes expected to be detected only due to pseudonym changes between two successive beacon emissions ($f = 10$ Hz).

saving efforts of developed PSC distribution strategies. Too high PSC inclusion frequencies lead to increased channel load increasing the probability of collisions on the wireless channel. Thus, the size of the covered communication area is decreased, which limits availability of information for applications. Hence, such behavior should be avoided.

The outlined VANET behavior is independent from whether a silent period is used after the pseudonym change or not. However, one should note that the length of the period during which awareness about the affected node by other nodes is poor is longer than the silent period itself. After the silent period, the PSC of the node has to be distributed as well to restore awareness.

4.4.2 Mix Zone based Pseudonym Change

The mix zone concept proposes to perform PSC changes within well defined geographical areas, which realize coordination between nodes for pseudonym changes. The minimum size of a mix zone depends on the accepted probability of an attacker being able to track a node although it changed its pseudonym within a mix zone. Thereby, larger mix zones significantly reduce the success rate of an attacker [251].

Prior work on the placement of mix zones has so far only considered privacy aspects. However, pseudonym change can significantly limit the capabilities of applications as shown in [205]. To keep such impact as small as possible, the areas affected by pseudonym changes should be kept as small as possible. A sketch of a mix zone together with the area affected by the pseudonym changes within the zone is given in Figure 4.8 to illustrate the issue of an additionally affected area around the mix zone itself. A common communication range in the whole scenario is assumed to determine the given sizes of affected areas.

One can see from Figure 4.8 that the affected area is larger than the mix zone itself. The core reason for this behavior is that once nodes leave the mix zone, they start to distribute their individual data sets, e.g., their PSC, anew. Thereby, every node leaving the mix zone is recognized as new a node by all other nodes (nodes within the directly affected area in Figure 4.8).

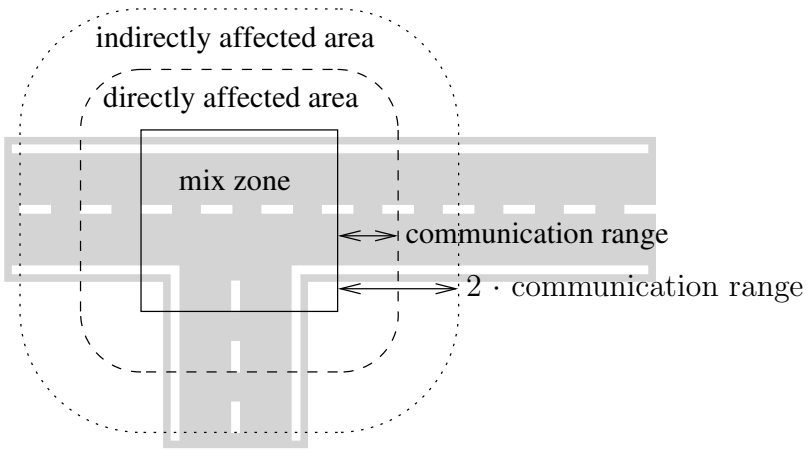


Figure 4.8: Sketch of a mix zone and the area affected by coordinated pseudonym changes.

Due to neighborhood aware PSC distribution, the area close to the border of the fixed mix zone will steadily experience increased channel load, in comparison to a scenario without a mix zone. Thus, nodes inside that area close to the mix zone will suffer from increased packet loss ratios and lowered communication radius (directly and indirectly affected areas in Figure 4.8).

In comparison to the case of uncoordinated PSC change, the VANET is only affected in parts from the presence of the cross influence of PSC distribution and change strategies. However, the size of areas with unreliable VANET performance is increased. Thus, methods to limit the influence of negative impact from PSC change on VANET performance are discussed in Section 6.5.

4.4.3 Ad-hoc Cooperative Pseudonym Change

During an ad-hoc initiated cooperative pseudonym change procedure, participants which are geographically close to each other change their PSC simultaneously. The location of the cooperative pseudonym change is found in an ad-hoc manner, in contrast to fixed mix zones. Moreover, not all nodes in vicinity have to take part. Different approaches on how to select cooperation partners have been proposed [251].

As for other PSC change techniques, evaluation has so far focused on the level of privacy obtained from the cooperative strategies. However, from an application's perspective it is also important to consider the time until cooperative awareness has recovered after a pseudonym change. This recovery time relates to a geographical area around the ad-hoc found pseudonym change area. The size of this extra area of lowered cooperative awareness clearly depends on communication conditions after message sending starts again at participating nodes. In case of poor communication conditions, e.g., high channel load, the recovery time will be increased. Thus, the area of poor cooperative awareness will increase, too.

For cooperative PSC change strategies, the locations of change areas are not fixed. Thus, the performance impact of PSC change is spread over the whole VANET, similar to the case of uncoordinated pseudonym changes. This can even increase the need for low recovery times, as

the performance impact can not be limited to well defined areas, i.e., well placed fixed mix zones. In case all nodes within a dedicated area take part in the PSC change process, the performance impact is equal to the one of a mix zone being present in that particular area, i.e., nodes set up an ad-hoc mix zone.

4.5 Summary about Overhead caused by Security Mechanisms

Several sources of security related overhead have been discussed in prior sections of this chapter. Moreover, they have been shown to significantly influence overall VANET performance. The main results of this chapter are,

1. the chosen type of platform independent data representation scheme can significantly influence the size of messages sent over the wireless channel, and binary XML data encoding provides shortest messages within ETSI ITS data sets,

2. PSC distribution is not fully specified in [125], and the combination of unsecured implicit and explicit requests with repeated explicit requests provides the best results in regard to cryptographic packet loss in comparison to more strict schemes,

3. the combination of a strict messages size limitation, lack of fragmentation support and uncoordinated variable content length on multiple layers leads to frequent violation of the access layer's message size limit, which can be overcome using a cross-layer content coordination scheme,

4. cross influence between pseudonym change and PSC distribution mechanisms threatens to (partly) disable the channel load reduction mechanism from of PSC emission strategies by superfluous new neighbor detections.

For more details on the individual topics the reader is referred to corresponding sections within this chapter. In the following section advanced attacks on VANETs are studied. Some of the found DOS weaknesses are caused by security related overhead discussed in this chapter. To overcome some of the weaknesses given above, advanced strategies for certificate handling are discussed in Chapter 6.

Chapter 5

Advanced Attacks on VANETs

An overview of attacks on VANETs proposed in prior work is given in Section 2.3. Several new attacks are suggested and evaluated in the following. These are specific realizations from the attack classes of

- DOS attacks (Sections 5.1, 5.2 and 5.3),
- de-pseudonymization attack (Section 5.4),
- Sybil attack (Section 5.3), and
- replay attack (Section 5.3).

Especially, the introduction of VoD and usage of GNSS based data sets are found to massively put security of VANET communication at risk. New requirements on VANET security mechanisms are derived from the found weaknesses, which restore secure and efficient communication under the presence of identified threats. These requirements are used to improve VANET protocols in later chapters of this work.

5.1 Denial of Service Attacks Misusing Protocol Functionality

The standardized certificate (chain) distribution mechanism of ETSI ITS and WAVE is described in Section 2.2.4. Two independent DOS style attacks on either PSC or certificate chain distribution are identified and evaluated in the following. At first, the attack on PSC dissemination based on misusing neighborhood aware PSC emission is discussed in Section 5.1.1. An attack on the explicit certificate chain emission request approach is introduced in Section 5.1.2. Both attack proposals assume a local, static, active outsider attacker. To maximize the impact caused by the attack, the chosen attacker always ignores all DCC rules.

5.1.1 Pseudonym Certificate Distribution

Implicit and explicit requests for PSCs can be carried out without knowledge about valid credentials. Messages, e.g. CAMs, without an included PSC can cause such requests [125, 176].

This feature helps to significantly speed up PSC distribution during ordinary VANET operation, as shown in Section 4.2.1. Topics discussed in the following are partly covered by prior work of the author in [32][1].

An active attacker can misuse the described PSC request mechanisms to cause bogus PSC emissions in the following way. The attacker sends out beacon messages with the maximum frequency used by valid nodes. For ETSI ITS and WAVE this frequency is 10 Hz. In doing so, the messages do never carry a PSC. Moreover, the hash value identifying the signer's PSC (i.e., the HashedId8 [125]) and the digital signature consist of random data. Due to the missing PSC, no receiver can check the signature, but will regard the message as an implicit PSC request, i.e., a new neighbor gets detected. Thus, all receivers will include their PSC in the next message. Rapid sending of the bogus request messages can cause receivers to always include their PSCs' in every message. Hence, the channel load is increased and communication conditions get worse.

The impact of the attack is illustrated in Figure 5.1 using the freeway scenario from Section 3.2. However, the basic mechanism of the attack is independent of the road topology in which the attack happens.

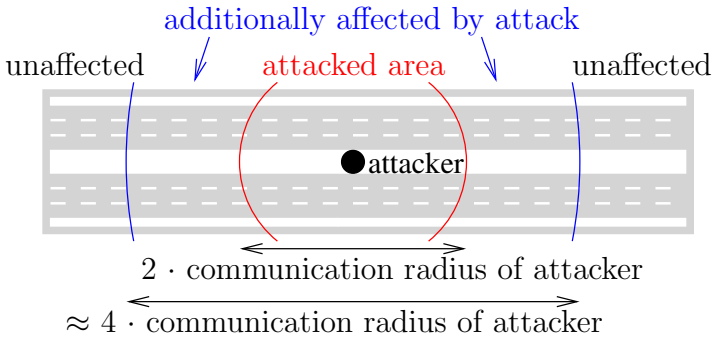


Figure 5.1: Sketch of the impact of the attack on PSC distribution in a freeway scenario.

Within the area having direct communication with the attacker (red area in Figure 5.1), nodes are affected by the attack in two ways. Firstly, they are caused to frequently send out their PSCs. Secondly, they suffer from increased channel load caused by themselves and the nodes in their environment. Nodes being within communication range of vehicles within the attacked area (blue area in Figure 5.1), only suffer from increased channel load, but do not add to the increased channel load themselves.

Valid nodes could use blacklisting to avoid responses to rapid requests from the same node. However, the attacker could just change the used sender identity after each sent message to circumvent such kind of countermeasures.

To evaluate the attack's impact on VANETs, the simulation environment from Section 3.3 is used. Results for channel loads experienced within the freeway scenario (see Section 3.2) in the area around a single static attacker are given in Figure 5.2. The given error bars show the obtained standard deviation of measured values, and the attacker is placed as illustrated in Figure 5.1.

[1] See also footnote 3.

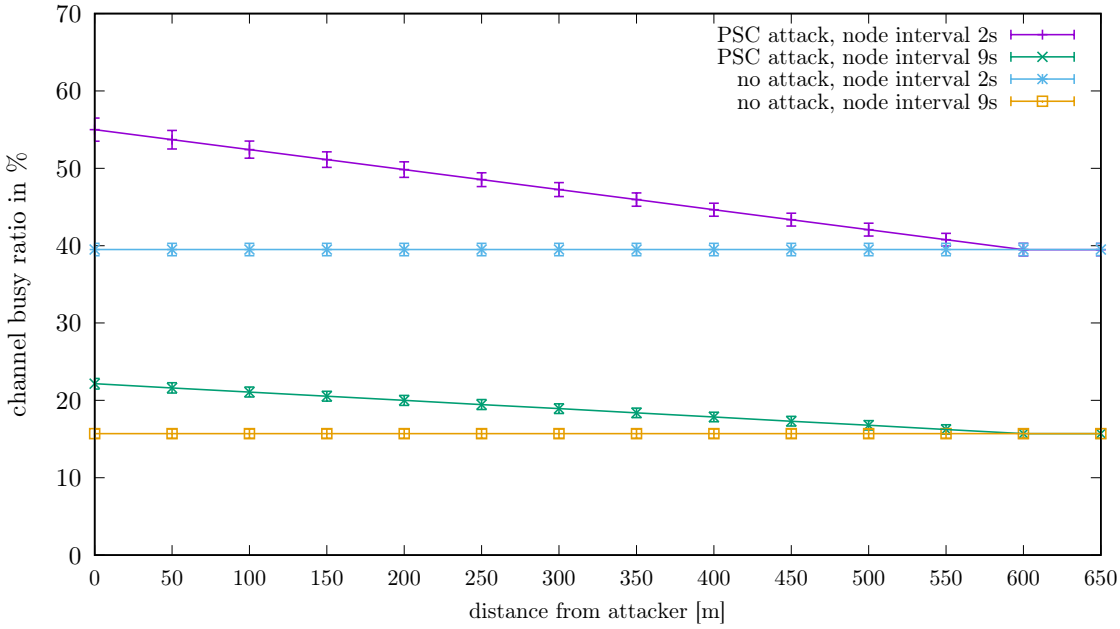


Figure 5.2: Impact of the attack on neighborhood aware PSC distribution.

To obtain the results from Figure 5.2, static, passive RSUs are placed in the area between the different driving directions of the scenario every 50 m. These nodes are only used to measure the channel load at their positions.

Results displayed in Figure 5.2 show that the attack is able to significantly increase the channel load around the attackers location. This holds especially for scenarios with high traffic density, i.e., with an average node interval of 2 s. Nodes close to the attacker almost share the same communication area with the attacker. Hence, these nodes are (almost) completely surrounded by nodes also being attacked. Thus, within their surrounding all nodes always include their PSC. In contrast, nodes at the edge of the communication area of the attacker receive about 50% of messages from nodes always including their PSC, while in the remaining messages ordinary PSC inclusion behavior happens.

Furthermore, the conducted experiments show that one requires to set up attackers at distances of about 300 m alongside the highway to cause all nodes in the scenario to always include their PSC. Thereby, the experienced channel load is equally high everywhere in the scenario.

5.1.2 Certificate Chain Distribution

Basic mechanisms for certificate chain distribution are outlined in Section 2.2.4. In contrast to PSC only distribution, the dissemination of certificate chains, i.e., an array of at least two certificates, uses only explicit requests [125]. However, both request schemes share the property that unauthenticated requests are permitted.

An active attacker issuing bogus certificate chain requests can massively increase the size of receivers messages. Within ETSI ITS two certificates (PSC and AAC) are included in the next

CAM after such a request got received [125]. However, the limited number of entries in the so-called certificate request list limits the amount of certificates, which an attacker can request in a single message. In current standards the maximum number of entries is six [125, 176]. The limited amount of certificates, which can be requested by a single message from the attacker, also limits the amount of receivers responding to the bogus request. However, the amount of different AACs can be expected to be low, e.g., one per OEM. Thus, many nodes share the same AAC. Moreover, the attacker can monitor the distribution of used AACs in his surrounding, by inspecting the PSCs of valid nodes. This allows him to always request the AACs being used at most within his sphere of influence. Thereby, the attacker maximizes the amount of responders, which also maximizes the caused channel load increase.

The attacker has two possibilities to cause emission of certificate chains by valid nodes. He can either

1. send out AAC requests himself (so called *direct attack*), which are part of beacon messages holding no PSC. All receivers using a requested AAC will respond with including their certificate chain. Alternatively, an attacker can

2. send out beacon messages holding a malicious PSC (so called *indirect attack*). This PSC identifies its issuing CA by a well chosen value in the signer ID field, which is different from all IDs of valid AACs. Thus, receivers will not know about the AAC and will transmit an AAC request themselves. However, the attacker chooses the ID in a way that its shortened version, which is used for the AAC requests by valid nodes [125], is equal to the one of a valid AAC, i.e., the attacker creates an address collision of the shortened IDs. Hence, the attacked valid nodes are caused to send requests for AACs being really present in the network. Therefore, all nodes using the requested AACs respond by emitting their certificate chain.

For the direct attack (case no. 1), the attacker can cause sending of six different certificate chains by a single malicious message. In contrast, the indirect attack (case no. 2) requires to send a dedicated message for each certificate chain.

To create the required AACs IDs for the indirect attack, the attacker just takes the shortened IDs of valid AACs and enlarges them by some random value with appropriate size. Rapid changing of the used fake AAC IDs helps the attacker to avoid blacklisting of its messages by valid nodes.

5.1.2.1 Evaluation

To evaluate the impact of the attack outlined above, the simulation environment described in Section 3.3 is parametrized with an AAC distribution following the OEMs' market shares within Germany from [198]. For all experiments the freeway scenario from Section 3.2 is used and the attacker is placed as shown in Figure 5.1. The error bars in all shown figures represent the standard deviation. The attacker tries to realize the DOS attack on nodes by increasing the channel load such far that data exchange between nodes becomes highly unreliable. Hence, the focus of this evaluation is on the channel load under presence of the attacks outlined above.

Results for the channel load in case of a direct DOS attack on AAC distribution are given in Figure 5.3. To obtain the given results, the attacker is placed like illustrated in Figure 5.5.

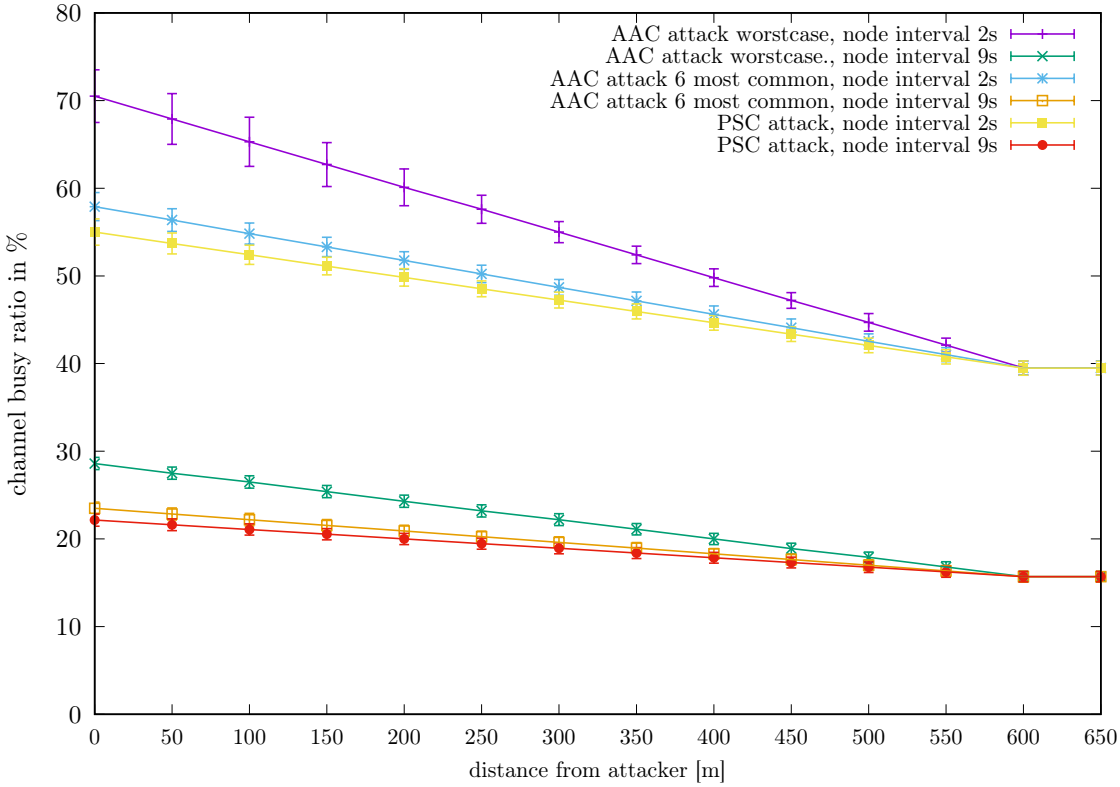


Figure 5.3: Impact of the direct DOS attack on AAC distribution.

Obtained results clearly show, that an active attacker can cause valid nodes to significantly increase the channel load. The effect is stronger for a higher traffic density. This is the expected behavior, as the single station attacker targets more nodes at the same time. Hence, more nodes take part in sending superfluous data to the single wireless channel.

Furthermore, even in case the attacker can only attack vehicles with the six most common AACs in his surrounding the impact is still significant. One can see that in case of an average node interval of 2 s, the CHBR is increased above the level of 40%. According to parameters from [103], this means that DCC is put into state RESTRICTIVE, while it is in state ACTIVE below this threshold. Hence, the CAM generation interval is restricted by the attack [119].

The geographical distribution of the impact of the direct attack is similar to the one of the PSC distribution attack from Section 5.1.1 illustrated in Figure 5.1. However, the channel load increase is much higher for the AAC distribution attack in comparison to the PSC distribution attack. This is caused by forced inclusion the entire certificate chain instead of only the PSC.

Evaluation results for the indirect attack are provided in Figure 5.4. The used attacker is positioned as shown in Figure 5.5. One can see from the comparison of Figures 5.3 and 5.4 that the extension of the attacked area, which is described above, actually happens. Moreover,

in contrast to the direct attack there is a significant area around the attacker in which an almost continuously very high level of channel load can be maintained by the attacker. The size of this area corresponds to the communication area of the attacker.

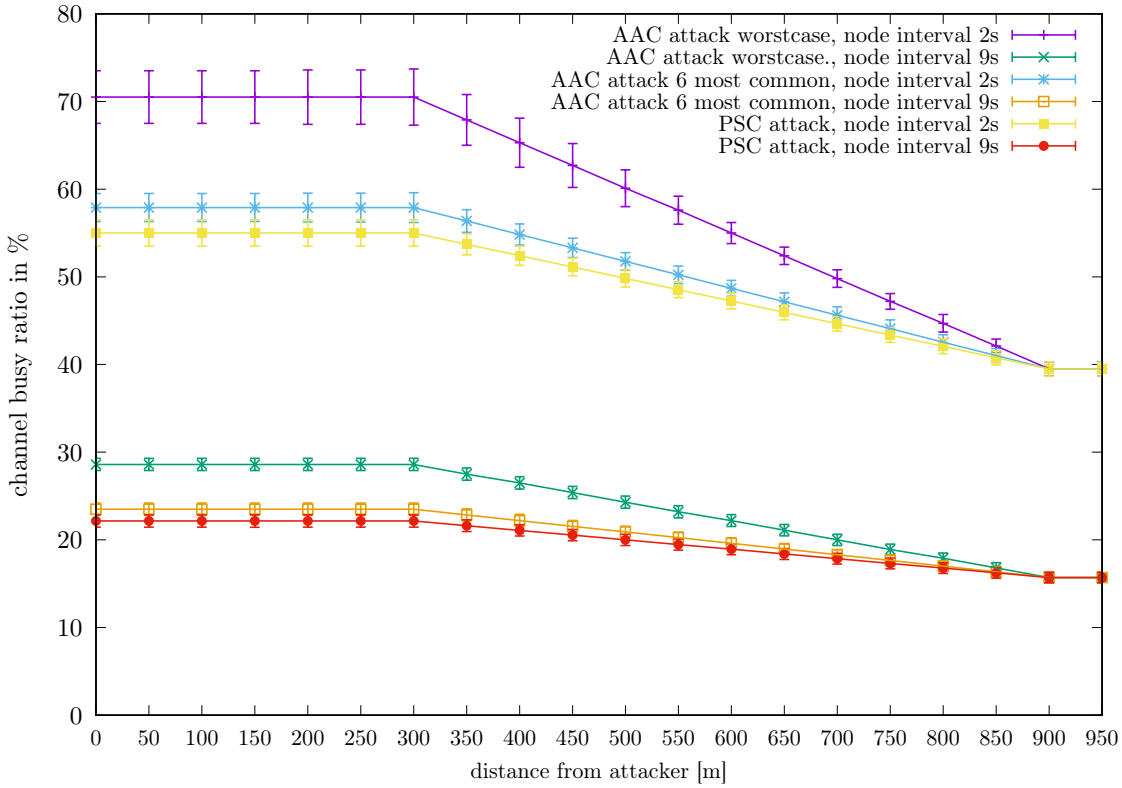


Figure 5.4: Impact of the indirect DOS attack on AAC distribution.

Outside the communication area of the attacker, the channel load decreases alongside with increasing distance to the border of that zone. This behavior is equal to the one shown in case of the direct attack starting directly at the location of the attacker. This shows that the valid nodes targeted by the indirect attack behave very similar to the attacker in case of the direct attack, i.e., they become involuntary co-attackers. The channel load increase disappears at about three times the communication range of the attacker.

The impact of the indirect attack on VANET channel load is also illustrated in Figure 5.5. All nodes within direct communication range of the attacker (dark red zone) transmit their certificate chain rapidly, but are additionally surrounded by other nodes showing that behavior, too. Moreover, these nodes rapidly send out AAC requests. Those requests are received within the dark red zone itself as well as within the neighboring red zone.

Moreover, there are nodes, which receive the caused AAC requests, but do not receive the malicious messages from the attacker (red zone without dark red zone in Figure 5.5). Hence, theses nodes transmit their certificate chain rapidly, but do not issue malicious AAC requests on their own. They receive messages from the dark red zone frequently holding a certificate chain,

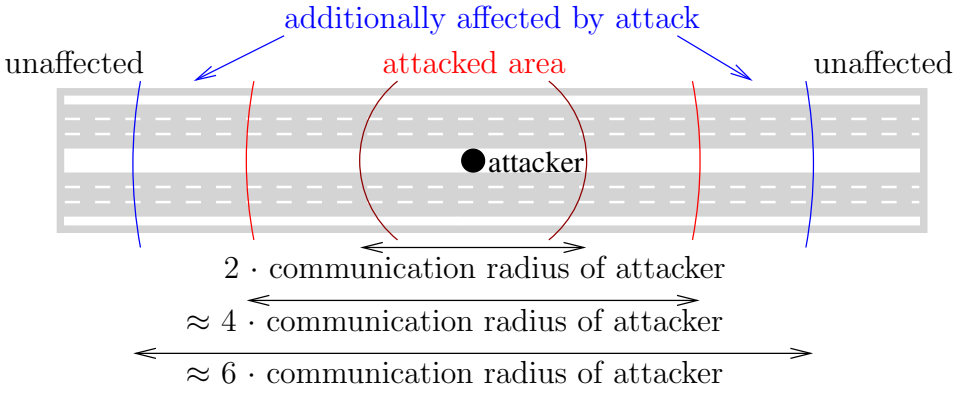


Figure 5.5: Areas around an attacker affected by the indirect attack on certificate chain dissemination.

but also from the blue zone (without the red zone). Furthermore, there are additionally affected nodes (blue zone without red zone), which do not add to the channel load increase themselves, but suffer from the increase caused by other nodes within the red zone.

5.1.2.2 Countermeasure

An effective countermeasure to the described attack would be to use a common AAC for all nodes. Thus, there would not be a requirement to distribute AACs on demand between nodes. Additionally, a common AAC would also help to improve privacy of nodes, as outlined in Section 5.4.

Another approach for significantly limiting the impact of the found attack is to limit the number of AAC deliveries, which follow an AAC request. Such behavior would also be beneficial for ordinary AAC distribution, by minimizing its impact on remaining VANET communication (see also Section 4.2.2). Thus, mechanisms allowing to limit the number of responses to an AAC request are studied in detail in Section 6.3.

5.2 Attacks on Verify-on-Demand Schemes

An introduction to the concept of VoD is given in Section 2.2.4.6. A set of attacks on VoD using VANET implementations is given in the following. These attacks have not been discovered in related work, except of prior work of the author in [40][2]. They assume a local static outsider attacker.

[2]Contribution of the co-author is mainly related to dedicated network and facility layer topics. Discussion of cross layer issues was done in close cooperation of both authors. Overall, the main contribution is from the author of this work, especially in regard to the proposed attacks.

5.2.1 Denial of Service Attacks by Misuse of GeoNetworking Features

VoD is based on the assumption that verified data is only required by applications. However, this only holds in case lower layers of the protocol stack act in a totally stateless manner, like in WAVE. This is not the case within ETSI ITS, as the network layer keeps a neighborhood table to provide the feature of message forwarding, i.e., multi-hop communication support. This leads to the two security vulnerabilities given in Sections 5.2.1.1 and 5.2.1.2.

5.2.1.1 Neighborhood Table Poisoning

The neighborhood table on the network layer is updated with each received message. In case messages are not verified before the update is performed, an attacker can poison the neighborhood table by

- adding extra bogus entries, and/or
- causing bogus updates of existing (valid) entries.

Incorrect entries in the neighborhood table can cause incorrect forwarding of messages (i.e., failure to forward or superfluous forwarding). With the used CBF method for forwarding in ETSI ITS, an attacker could fake the position of a multi-hop message's sender in a way to make valid forwarder candidates not forward the message. Thus, the attacker can perform a DOS attack on multi-hop communication.

Multiple countermeasures to the found weakness can be thought of including

- verification of all messages before the neighborhood table update happens. However, this completely disables VoD, as every received message gets verified.
- Instead of replacing entries in the neighborhood table, one could keep prior entries, too. Old entries are only removed after a later update got verified by another mechanism. However, this significantly increases memory requirements, due to an expected low number of verifications.
- One could only store entries in the neighborhood table after the corresponding message got verified. However, low numbers of verifications will cause neighborhood tables to be (very) sparse. Thus, it can be expected that routing will significantly suffer from this approach.

The found disadvantages of countermeasures together with the impact of the attack itself lead to the conclusion that the combination of approaches requiring neighborhood table keeping with VoD is not recommended.

5.2.1.2 Denial of Service Attack by bogus Multi-Hop Messages

ETSI ITS standard [122] requires a received multi-hop message to be verified in case the receiver is about to forward the message. This is done in order to prevent an attacker from flooding

the VANET with bogus messages and creating harmful channel load in a large area, as the dissemination area of a multi-hop message is not limited in general.

[54] suggests to limit the size of dissemination areas to several kilometers, but this still massively exceeds the communication range of the attacker. Hence, unverified message forwarding would still significantly increase message injection capabilities of attackers.

The used forwarding strategy in ETSI ITS is CBF. Thus, forwarder selection is done in a decentralized manner. A node considers itself a forwarder candidate in case it can achieve progress towards the destination. This decision is done by comparison of the own position and the position of the sender, which is obtained by a look-up in the neighborhood table based on the sender's MAC address. Afterwards, the timeout for sending is started, as described in Section 2.1.2.

To ensure that targeted nodes consider themselves forwarder candidates, the attacker hijacks a valid identity of a node present in the VANET. He can easily obtain such identities from receiving valid CAMs. An example for such an attack is given in Figure 5.6. In the given example, the attacker sends out two bogus messages. One is claimed to originate from node A and one from node B. Node C is caused to trigger verification of both bogus messages, while the remaining nodes only consider themselves forwarder candidates for one bogus message (either claimed from node A or B).

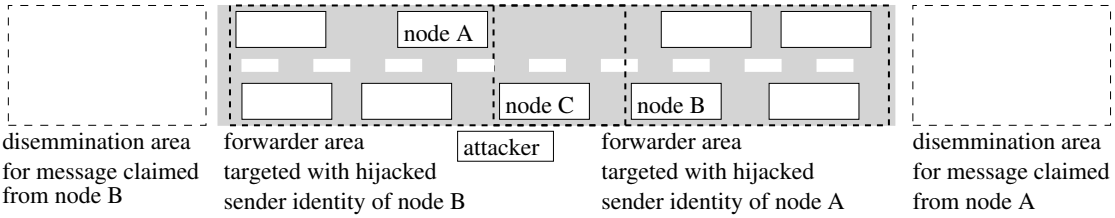


Figure 5.6: Example for DOS attack on VoD by bogus multi-hop messages.

Two possibilities exist for when to trigger verification of the to be forwarded message. Either verification is triggered when the timeout is started or after the timeout has elapsed. The first variant provides the advantage of parallel usage of the timeout time for verification and forwarder selection. However, this strategy causes all nodes which consider itself forwarder candidates to verify the to be forwarded (bogus or valid) message.

The core advantage of CBF over sender based forwarder selection is to use up to date information about the current distribution of nodes in the network. Thus, the length of the timeout interval before message forwarding is selected to be small [141]. The verification delay would increase that delay after the sending timeout has already elapsed. Thus, during verification the knowledge used for forwarder selection will become somewhat out of date. Hence, verification should be fast, which is somehow in contrast to the aim of VoD targeting to reduce performance requirements for verifications. Moreover, this strategy cannot avoid verification of bogus messages at all forwarder candidates. The attacker cannot correctly sign the message, which causes the verification to fail. Thus, the messages never gets forwarded. Hence, the timeout at all forwarder candidates will elapse (without being canceled) causing message forwarding, i.e., verification.

However, there is an advantage for the case of valid messages. During the time for verification, the sender timeouts of additional nodes may elapse. This still causes message verification at multiple nodes. However, as long as the verification delay is smaller then the maximum timeout interval there is the chance to spare verifications at nodes, which are not going to forward the message at first. Therefore, verification after the CBF timeout time is recommended for nodes providing significantly fast signature verification.

The outlined attack can be easily targeted to a single node in case sender based forwarder selection is used. Thereby, the attacker can explicitly select the targeted node as the forwarder. However, an attacker has to target every node individually, while attacking all nodes at once is possible for the case of CBF-based forwarding.

In case the verification capabilities of a node are highly limited, an attacker can exceed such capabilities by just sending more multi-hop messages to a node then can be verified. Thus, verification of all kinds of messages is (massively) delayed and if storage capabilities of corresponding buffers are overwhelmed affected messages will have to be dropped. Hence, such messages cannot be used leading to failure to forward or unavailability of information for ADAS. This clearly makes the attacker achieve his aim of performing a DOS attack, not only on multi-hop communication, but on the whole VANET input of affected nodes.

One way to avoid the outlined attack is to be able to verify all incoming messages. However, this contradicts the VoD aim of limiting the performance requirement for message verification. An alternative countermeasure is to use prioritization of verifications. Thereby, one can only perform them for to be forwarded messages in case the single hop messages leave enough spare verification capacity. While this would allow an attacker to perform a DOS attack on the multi-hop part of communication, single hop communication (like dissemination of CAMs) still works under presence of the attack somehow limiting its impact.

5.2.2 Denial of Service Attack by bogus Triggering of Applications

VoD triggers verification of VANET messages in case these would be used by ADAS to perform a safety critical operation, e.g., to display a warning message to the driver. From the definition of ADAS, e.g., in [111, 112], an attacker can easily determine the condition a bogus message has to fulfill to be regarded as relevant by an ADAS. Straight forward examples include all kinds of road hazard warnings (like icy road warning), which are taken into regard by any node within the relevance area of the warning. The attacker can freely chose this warning area, to attack as many nodes as he wants to attack.

The impact of the attack is similar to the one of the attack on message forwarding outlined in Section 5.2.1. In case too many bogus messages have to be verified, proper verification of valid messages is at risk. This clearly limits the data quality available for ADAS. In case the attacker can send enough bogus messages to avoid verification of valid messages at all, he achieves a full scale DOS attack.

The only way to avoid dropping of valid messages' verifications is to be able to verify all received messages. However, this contradicts the aim of VoD, which is to limit the required verification performance. This shows that one can either choose to use VoD at the cost of limited system robustness, or to use a verify-all scheme to successfully avoid the outlined DOS weaknesses.

5.2.3 Attacks on Complex Data Processing on Higher Protocol Layers

A general method for keeping a system secure is to keep the interface(s), which are exposed to an attacker, as small as possible. For the case of VANETs, prior work has argued in favor of a change of the security envelope's format, to avoid parsing of its content before signature verification takes place [237]. A verify-all scheme only exposes low level data processing interfaces up to the network layer security entity.

In contrast, the VoD concept suggests to parse the whole message on all protocol layers, before deciding on whether to verity the message at all [199]. Thus, the surface for an attack on data parsing and usage is significantly increased by VoD in comparison to a verify-all scheme.

Within ETSI ITS the data sets on several protocol layers use more complex data encoding schemes in comparison to WAVE, for which VoD was initially proposed. ETSI ITS protocol layers above the MAC layer use variable length data sets and deeply nested data types [119, 122, 125]. On the facility layer ASN.1 encoding, e.g., in the UPER variant for CAM and DENM, is used. Parsing of data encoded with such complex schemes requires complex implementations, which leads to a high risk of security problems. Even for very simple ASN.1 schemes, like the BER variant, many security problems have been found in implementations in the past [55, 70, 172, 173, 227, 323], e.g., the BERserk vulnerability [172, 173].

Therefore, the effort for secure implementation of all data processing units within a node handling received data is significantly increased by using VoD in comparison to a verify-all scheme. This finding puts the VoD concept into question from a system design perspective.

Overall, the found weaknesses of VoD lead to the conclusion that usage of this verification scheme is not recommended for usage within ETSI ITS. Instead, a verify-all scheme should be applied.

5.3 GNSS Spoofing based Attacks on VANETs

GNSS spoofing attacks are a well known problem, especially for GPS, as outlined in Section 2.5.2. However, there is a lack of prior work dedicated to an analysis of GNSS spoofing targeting VANETs. This holds especially for the case of manipulated time information within the GNSS signal. As outlined in Sections 2.1.1 and 2.2.1, knowledge of accurate and reliable time and position information is fundamental to achieve a secure VANET system. VANET requirements for time synchronization are rather strict, e.g., a maximum deviation from the reference time of 20 ms is specified in [54].

The general outline and impact of the proposed attack on GNSS input of VANETs is given in Section 5.3.1. Countermeasures to the found security problems are discussed in Section 5.3.2. An experimental evaluation of the proposed attack applied to a commercial state of the art OBU is provided in Section 5.3.3. Topics covered in this section are partly covered by prior work of the author in [29, 37][3].

[3] Contribution of co-authors mainly relates to dedicated GNSS topics as well as to the design and carrying out of the experimental evaluation of the proposed attacks. The main contribution is from the author of this work, especially in regard to the found attack weaknesses and suggested countermeasures.

5.3.1 General Attack Outline

We consider two different attacker models. These are a

1. simple attacker, who is an active, local, outsider attacker. This attacker has no physical control over the attacked node(s), i.e., he can only carry out the attack over the air, and an

2. advanced attacker, who is an active, local, insider attacker. The advanced attacker has full control over a valid node being used to attack other nodes. This means especially, that he can manipulate in-vehicle information and switch the vehicle on and off. However, the assumed advanced attacker cannot access the sensitive key material inside the OBU directly, i.e., he cannot circumvent the protection mechanisms of the Hardware Security Module (HSM) holding the sensitive cryptographic material, e.g., private keys for PSCs.

In both cases, the attacker can send GNSS signals to the attacked node containing arbitrary time and location information. As outlined in Section 2.1.2, time and location information inside a VANET protocol stack is highly dependent on GNSS input. The contained time signal is used for time synchronization among nodes. Furthermore, absolute location information is obtained from GNSS. Thus, manipulation of these basic data sets can be expected to show a significant impact on the security of a VANET.

For the case of the simple attacker, the attacked vehicle is controlled by a valid VANET user, i.e., a driver who does not act as an attacker. The attacker tries to attack VANET services inside the vehicle from the outside.

In contrast, the advanced attacker controls a vehicle whose OBU is attacked to generate malicious messages to be used for a (later) attack on valid nodes. In doing so, the attacker makes use of the valid credentials of the node he controls. One should note, that an attacker can try to hide his identity in case of a detected attack, but not the identity of the used node, as AAs can link the used pseudonyms to the node. However, vehicle lender's fleets or car sharing fleets are vulnerable to the described attack, as the time of the actual attack on the VANET is (almost) independent from the time of controlling the misused valid node.

5.3.1.1 Impact on Security Functionality

Time information is used within the security entity for two main purposes [125], which are

1. setting or checking the sending time stamp of a message, and

2. checking the validity time restriction of certificates.

Thus, an attacker manipulating the time information can make an attacked node emit messages with any time stamp, for which a valid certificate chain is stored. Moreover, all received messages valid at the point of time set by the attacker are accepted.

Absolute position information is used within the security entity for two main purposes similar to time information [125]. These are

1. setting or checking the sending location stamp of a message, and

2. checking the geographical validity restriction of certificates.

Hence, an attacker controlling the location information can make an attacked node transmit messages with any location stamp, for which a valid certificate chain is present. Furthermore, all received messages valid at the location provided by the attacker are accepted. One should note that geographical validity restrictions are not foreseen for PSCs in ETSI ITS. Moreover, the security envelope of CAMs do not hold a location stamp in contrast to other kinds of messages [125]. Thus, manipulation of the location component is not required for CAM based attacks.

Location spoofing cannot be used against an RSU, as it has a fixed location. Thus, this location can be stored during setup of the station and no corresponding updates are required during its operation.

In case an attacker targets only a single isolated node, the neighborhood table of this node will be empty. Thus, the station will not send CAMs but only pure beacons. The attacker can easily change this, by transmitting own beacons to the targeted node. Current ETSI ITS standards do not require beacons to be signed by the security entity [122]. Therefore, the attacker does not need access to valid key material to generate the required beacons. This is clearly not required in case the attacker can target at least two vehicles, which will mutually initiate the transmission of CAMs once they recognized each others beacons.

In the following, a number of different attacks is described, which are enabled by successful GNSS spoofing of VANET nodes.

DOS Attack Temporal validity restrictions of PSCs enable a simple attacker to perform a DOS attack on nodes. To carry out the attack, the internal time of the attacked node(s) is set to a point in time (past or future) for which they do not hold a valid PSC and also PSCs of nodes within their neighborhoods have either passed the end of their lifetime or their lifetime has not started yet. This causes two effects, which are

- an attacked node is not able to send out any further VANET message, as there is no valid PSC to sign it, and
- an attacked node will discard all received messages, as they seem to be signed by certificates being used outside of their lifetime. Moreover, messages either seem to be massively outdated or to come from the future, which also causes their discarding [54].

Thus, the attacker can ban any further communication between the attacked node and the remaining VANET, which leads to a successful DOS attack. An analogous attack can be performed misusing geographical restrictions of PSCs, too. Unlike for CAMs, the security envelopes of BSMs also use locations stamps and not only time stamps.

Acceptance of Outdated Messages An attacker can receive and store valid messages. The simple attacker just uses the messages of nodes he cannot control. In contrast, the advanced attacker can drive arbitrary trajectories and store the corresponding VANET messages emitted by the node under his control.

After having recorded the data sets, the attacker sends out the stored messages in a replay attack. Moreover, he transmits a faked GNSS signal, which causes the time inside attacked

nodes to be in line with the time stamps of the replayed messages. This makes the attacked nodes accept the replayed messages. Thereby, bogus virtual nodes can be suggested to various protocol stack entities. For example, the neighborhood table on the network layer will store the invalid nodes as possible message forwarders. Misbehavior of applications by reaction to the presence of the invalid nodes may be caused, too.

Acceptance of Outdated Certificates Similar to the case of outdated messages, an attacker can cause the acceptance of outdated certificates by resetting the internal time of targeted nodes into the past. In case of pure replay attack, this means that in addition to the validity time check of a message also the corresponding validity checks of certificates used to secure the messages are passed. Thus, the outdated message is accepted as a valid one by the receiver.

Moreover, acceptance of outdated PSCs is a particular sensitive issue in regard to access control in VANETs. In case an attacker can get access to formerly valid, but outdated sensitive key material he can use the GNSS-based attack on internal time of valid nodes to circumvent the access control feature. The attacker can use the outdated key material to generate arbitrary messages and inject them into the VANET. These messages will be accepted by valid nodes as outlined before.

Even in case such an attack gets detected, there is currently no mechanism in ETSI ITS to ban such an attacker from accessing the VANET, as there are no Certificate Revocation Lists (CRLs) in ETSI ITS. Instead, nodes with detected misbehavior do not receive PSC updates from AAs. This approach is intended to make such nodes run out of valid PSCs over time. However, this mechanism does not work in case the outlined GNSS attack is used, as the misbehaving nodes can just use their outdated certificates. In contrast, CRLs are used in WAVE [176]. Hence, the problem affects ETSI ITS in a more severe way than WAVE.

Acceptance of Messages Outside Validity Range Location stamps within the security envelope are used to limit the validity of a message to a dedicated area in connection with a geographical validity restriction of the corresponding PSC. ETSI ITS uses this kind of restriction for DENMs [125].

An attacked node using manipulated location information can be caused to accept messages, which are distributed outside their region of validity. The malicious message can originate from a replay attack.

Creation of Messages with Future Time Stamps To generate signed messages with future time stamps, an attacker sends a spoofed GNSS signal with target time t_f (in the future) to the OBUs of attacked nodes. The advanced attacker can clearly start the attack before starting up the car. Thereby, the GNSS receiver receives the manipulated signal from begin of its operation on. This makes the attack more probable to succeed as the receiver's possibilities to detect the attack are greatly limited [299].

The point in time most far in the future the attacker can use for t_f is the end of lifetime of the PSC being valid for the longest time from current time on. This point in time is denoted by $t_{f,max} \geq t_f$.

If t_f is sufficiently far in the future (i.e., the attacker has enough time for carrying out the attack) the attacker can repeat the procedure described above again and again until he has obtained faked messages signed by all PSCs contained in the OBU, which are valid at t_f. Thereby, a successful Sybil attack [86] can be performed, as the attacker can use multiple well signed sets of messages in parallel. This weakness especially affects VANET implementations using a so called certificate pool, i.e., many PSCs with overlapping validity time spans are stored within an OBU to keep the frequency of certificate refill procedures low. The advanced attacker can easily create multiple sets of messages from the same time range using multiple stored PSCs with overlapping validity time periods. To do so, he just switches the vehicle off and on again causing a re-start of the OBU. This causes selection of a new PSC (see e.g., [54]).

Even in case the attacker can not directly control a vehicle's start up, he can still use the described attack once on every node within the range of his manipulated GNSS signal to obtain properly signed messages from the future. This clearly violates the VANET system security requirement of non-repudiation. In case a start up of a targeted car is required, an attacker can target places with high numbers of such procedures happening, like car parks.

One should note that this kind of attack is especially serious for vehicles with rapidly changing users, e.g., those from car sharing or car lenders fleets. An attacker can temporarily use a vehicle from the fleet and generate future messages with its PSCs. Afterwards, he uses the generated and recorded messages (e.g., CAMs and DENMs) significantly after he returned the vehicle. Even in case the nodes misbehavior is detected, the vehicle's user at the time the attacker performed his replay attack will be suspected of having caused the misbehavior. This is due to the expected non-repudiation property of the security system, which was actually circumvented by the attacker.

The only kind of validity restriction of certificates not affected by the GNSS attack is the limitation of a certificates to a dedicated set of ITS-AIDs, and corresponding SSPs. These kind of usage limitations only relate to the granted capabilities of a node, but not to time or location related information.

5.3.1.2 Impact on Trajectory Modeling

Collision avoidance applications require quite detailed trajectory modeling of nodes to detect possible future collisions [285]. With the outlined attack, an attacker can manipulate the time and location data sets contained in application layer messages, e.g., CAMs and BSMs. These data sets are used by receivers to model the trajectory of the sender. Thus, the attacker can create a node with an arbitrary trajectory at receivers. In case no further validation of the input data can be performed, e.g., by using additional sensors like radar sensors, inappropriate reaction of ADAS may be caused by the attacker.

Moreover, multi-hop communication uses geographic routing in VANETs. To enable such routing, the absolute position of nearby nodes is stored in a neighborhood table inside each node together with a time stamp. This means that the performance of multi-hop communication depends on accurate trajectory modeling. Hence, manipulated GNSS data can be expected to decrease the performance of multi-hop communication.

5.3.2 Countermeasures

In general there are two types of countermeasures to the attack outlined in Section 5.3.1. One can either try to avoid that manipulated time and position data is used within the protocol stack, or the impact of such incorrect data sets can be limited to a minimum.

To avoid usage of incorrect time and position data within the protocol stack, detection of the manipulation is required. Existing generic countermeasures to GNSS spoofing can be applied. Such mechanisms try to detect the spoofing and disable usage of the manipulated data. This allows to (temporarily) disable the time synchronization and/or global position updates based on GNSS. Local sensors can be used to provide an estimate for the required data sets during the time of disabled GNSS input. For time data a local clock can be used, and a global position estimate can be achieved, e.g., by an inertial navigation system [261, 301] within the vehicle.

A simple context logic based approach to secure an OBU against invalid time stamp jumps is considered in the following (Section 5.3.2.1). Moreover, three different approaches for detecting a GNSS spoofing attack based on cross checks with other independent data sources are studied. These include cooperation between the VANETs' nodes themselves (Section 5.3.2.3), alternative time synchronization mechanisms (Section 5.3.2.4) and alternatives for absolute localization (Section 5.3.2.5). Furthermore, Section 5.3.2.6 describes a general method for limiting the impact of the found attacks by usage of short lived PSCs.

5.3.2.1 One Directional Time Stamp Jumps

Typical time synchronization systems, e.g., ntpd, use the basic concept that modifications of the time stamp can only occur in one direction, which is into the future. Thereby, the system time is ensured to be monotonically increasing. This strategy simply stores the highest value of the system's time stamp counter, and allows only higher values to be set afterwards. Within ETSI ITS, the security functionality is to be implemented in a temper proof way, i.e., inside an HSM. Moreover, the security entity needs to have access to the system time for embedding the time stamp into the security envelope [125]. Hence, the security entity could store the last (and highest) time stamp inside its tamper proof memory. This stored value can be used for a consistency check in case a jump in the reference time is detected. Thereby, resetting of the time base to past values to force acceptance of outdated messages and certificates can be avoided.

However, such an approach would open the possibility of a permanent DOS attack on nodes. An attacker could manipulate the time base of an OBU once, setting its system time very far into the future. Afterwards, the OBU cannot send out any kind of messages, as all its certificates are considered being invalid. They seem to be used after their lifetime ended. Additionally, the node will not be able to receive any valid messages from other nodes, as these seem to carry time stamps from the far past. Therefore, the security entity will drop all received messages. Furthermore, the certificates of other nodes will be regarded as invalid, as they are considered to be used after their lifetime ended. In case the time stamp cannot be reset to the correct value after the attack, the inability to communicate will be permanent for the attacked node. Hence, the attacker managed to perform a successful and very long lasting DOS attack on the node, which even persists after the attacker has long ended his active attack.

Unfortunately, the simple way of securing time stamp usage by forcing time stamp modifi-

cations to only occur into the future is not recommended, due to the outlined permanent DOS attack weakness.

5.3.2.2 Absolute Lower Time Stamp Limit

To limit an attackers capabilities for time stamp manipulation into the past, an absolute lower limit t_0 for the current time stamp t_c can be defined. In case a time stamp t_r is received from the reference clock for which $t_r < t_0$ holds, an attack is detected. Hence, such a t_r is not used for time synchronization of t_c. A proper initial value for t_0 can be set a manufacturing time.

Furthermore, a mechanism to cyclically update t_0 is desirable, to keep the difference between t_c and t_0 small. This limits the time span in which an attacker can manipulate t_c into the past. One possibility to realize such kind of mechanism is to re-use the time stamps of AA's PSC delivery messages to update t_0. Moreover, a new value of t_0 is only accepted in case it is higher, i.e., more in the future, than the currently stored value of t_0. This approach assumes that AAs have always access to a valid time source, e.g., by using their own atomic clock.

By employing the suggested mechanisms, the attackers capabilities can be limited to only manipulate t_c in the area of $t_c > t_0$. Hence, no acceptance of outdated messages or certificates with validity periods in the past before t_0 can be caused by the attacker. This clearly limits the capabilities of an attacker. However, practical impact depends on the t_0 update interval, which is bound to the PSC update interval in case the approach outlined before gets used. More rapid updates lower the size of time span an attacker can target, but also increases dependency on frequent backbone connections, which is undesirable from a VANET design perspective following ETSI ITS and WAVE like approaches.

5.3.2.3 Cooperative Time and Location Validation

So called cooperative localization is a well studied subject in wireless networks [226, 329]. It can be used to improve location accuracy at nodes and also to secure the location information. In doing so, each node performs cross checks of its own position estimate with the positions received within messages, e.g., beacons, from other nodes. In case large deviations are detected, an attack is assumed. This procedure could be extended to time stamps as well.

However, the described way of cooperative attack detection should not be used in VANETs. It would cause a severe DOS weakness in these systems. An attacker can easily record valid messages (CAMs or BSMs) and inject them into the system later on, as a replay attack. In doing so, the replayed messages can be selected to significantly differ from valid up to date messages in regard to included time and location stamps. However, their signatures and PSCs are still valid. Hence, valid nodes would detect an attack on time synchronization and/or location information, which causes incorrect disabling of the usage of this data sources.

5.3.2.4 Alternative Time Sources

An alternative to spoofing detection is to use independent time sources in order to perform mutual cross checks for detecting an attack on one of them. Such detection can compare the

difference between the last time stamp, which is considered valid, and its update to a well defined threshold. Candidates for alternative time sources include

- a local clock,
- terrestrial time broadcast,
- mobile phone network's dedicated time broadcast messages [113],
- 802.11p time announcement frames [169],
- Internet based time servers (e.g., via Network Time Protocol (NTP)).

These individual time sources are discussed in the following sections.

Local Clock An available hardware clock is a common feature of standard PC hardware. It can be used to determine the current time without assistance of any external entity [71]. However, realization of a reliable embedded absolute time source within an OBU faces several challenges.

An automotive ECU, like an OBU, needs to use a highly efficient power save mode quickly after engine shutdown [162]. This is required to avoid draining a vehicle's main battery. Furthermore, power disruptions have to be tolerated, e.g., in case the main battery is temporarily disconnected for performing vehicle maintenance. Usage of an extra rechargeable battery inside the OBU itself would significantly increase its price. Such kind of battery has to be able to run the time source for a period of at least some weeks without recharging, while tolerating tough environmental conditions.

Moreover, local time sources are subject to drifting problems [71], which is the reason behind synchronization to a more stable reference clock. Drifting issues are especially relevant in case of tough environmental conditions, as they are to be faced within the automotive domain [182]. This especially relates to significant temperature changes experienced by the clock [71, 212]. Typical local clock implementations utilize a quartz oscillator to derive time stamp increase steps. Such oscillators show a typical time drift of 10^{-6} $\frac{\text{seconds}}{\text{second}}$ to 10^{-8} $\frac{\text{seconds}}{\text{second}}$, while operated under about constant temperature [71]. Specified accuracy of implementations within the full automotive temperature range is typically worse, e.g., $3 \cdot 10^{-6}$ $\frac{\text{seconds}}{\text{second}}$ for a temperature range of -40°C - 85°C [238]. More accurate clocks based on rubidium or cesium oscillators are available in general. However, they are usually to expensive and not optimized for usage under automotive conditions.

Considering the requirement to keep the time source within an accuracy range of 20 ms in comparison to Coordinated Universal Time (UTC) [54], this means that a local time source with a drift of $3 \cdot 10^{-6}$ $\frac{\text{seconds}}{\text{second}}$ is expected to violate the time synchronization requirement after $\frac{20 \cdot 10^{-3} s}{3 \cdot 10^{-6}} \approx 6667$ s ≈ 111 minutes. Hence, quite shortly after a vehicle shutdown initial time synchronization is required before VANET communication with accurately enough time stamps can be continued after the next startup. Furthermore, even in case the GNSS attack gets detected and time synchronization is disabled, the need to re-initialize the time synchronization appears, as the local clock can only provide an accurate enough time base for a limited amount of time. Hence, in case the attacker can maintain the attack longer than that time span, the attacker's

target can only either generate messages with non-accurate time stamps or disable VANET communication, which leads to a successful DOS attack.

Moreover, slow but yet continuous deviation of the local clock from UTC is possible by GNSS spoofing, as some deviation between the received reference signal and the local clock has to tolerated by the local time synchronization algorithm. Such an attack is similar to the one shown in [298] for positioning, i.e., inaccuracy of local sensors is exploited to undetectably provide a spoofed GNSS signal to the receiver. Increased deviation accepted by the receiver leads to faster violation of time synchronization requirements. This shows that a local clock is hardly able to avoid the vulnerability of VANET time synchronization to the found GNSS attack, especially for the case of vehicles being attacked at their startup time, i.e., during initial time synchronization to UTC.

Every possibility to reset the internal state of the secondary time source allows an advanced attacker to perform the attacks given in Section 5.3.1. Therefore, one would have to include the time source into a HSM. This not only increases the costs of the HSM, but also means that secure re-initialization of the HSM is required each time its power supply got disconnected. This would probably cause significant overhead, e.g., during vehicle maintenance. Hence, extra secure in-vehicle time sources seem infeasible to overcome the described weakness of time synchronization within VANETs.

Terrestrial Time Broadcast Dedicated terrestrial senders of a time synchronization signal distributing legal time are available in many countries, e.g., with DCF77 in Germany or WWVB in the US [99, 132, 213, 252, 253]. These senders transmit a far reaching signal at pretty low frequencies allowing accurate time synchronization with low effort [236].

Unfortunately, terrestrial time broadcast is unsecured, i.e., messages do not carry any authentication or integrity information. However, they provide a second time source which is independent from GNSS [253]. Thus, using this input an attacker would need to spoof it together with the GNSS signal to perform the attacks outlined above.

Prior work on spoofing of terrestrial time broadcast is very limited. It was shown that standard PC equipment can be used to spoof the amplitude modulated part of the signal emitted from DCF77 [49]. However, no full attack on the entire signal has been published yet.

Due to the low amount of sender's, terrestrial time broadcast is a promising candidate for extension towards a secured time broadcast system. Moreover, governments have pushed forward VANET approaches in the past (see e.g., [154]). Thus, a security improvement of government operated time senders would fit this move towards increasing traffic safety.

Mobile Phone Network Time Synchronization The so called Network Identity and Time Zone (NITZ) message can optionally provide time information in mobile phone networks [113]. Unfortunately, distribution of this message is optional, frequency of distribution is not standardized and accuracy of the included time information is just in the order of five minutes. Thus, the message cannot fulfill time synchronization requirements for VANETs in its current form. Moreover, many operators of mobile phone networks do not distribute the NITZ messages within their networks. An unofficial list of operators utilizing this kind of message can be found in [326].

Moreover, mobile phone network stations often obtain their time base from GNSS [296]. Hence, they are also possible targets of a GNSS spoofing attack. Furthermore, security of NITZ messages' content is called into question by findings in various prior work, which showed that message injection is possible in mobile phone networks [11, 245, 249, 286]. Hence, securing VANET time synchronization via NITZ messages is not recommended.

Time Announcement Frames The 802.11p standard specifies optional so called time announcement frames to be used for time synchronization among nodes [169]. However, time synchronization using these messages is subject to two major drawbacks, which are

1. time announcement frames are sent by the MAC layer. Hence, they are not secured, as securing of messages within WAVE (and ETSI ITS) happens at the network layer. Thus, an attacker could fake time announcement frames.

2. There is no guarantee that originators of such frames have access to a valid time source. In contrast, wireless GNSS spoofing typically affects an entire area. Thus, an attacker can spoof the GNSS input of many or even all nodes exchanging time announcement frames. Hence, such frames do not provide independent time information to nodes.

Moreover, ITS-G5 does not make use of time announcement frames. Thus, they are not used in the current European VANET approach. Therefore, usage of time announcement frames is not a suitable countermeasure to the attacks outlined before.

Internet based Time Servers In case nodes have regular Internet access, well known NTP can be used for time synchronization between nodes. NTP implementations are provided by ntpd and chrony [2, 73]. However, NTP was found to be subject to multiple security flaws [217, 269, 297]. Thus, extra mechanisms than pure provision of access to an Internet based time server are required to allow nodes to obtain time synchronization information in a secure way.

One way to ensure avoiding the outlined security weaknesses is to provide the time server via a Virtual Private Network (VPN) [72, 289] to a secured backbone service. In this case, one must ensure that the backbone does not obtain its time from GNSS. Otherwise, an attack on the time server based on GNSS spoofing would affect all nodes being assigned to that server. A disadvantage of such a solution is to introduce the requirement of rapid backbone connections for all nodes, which also leads to high computational requirements for the backbone service. Requirements for secure time synchronization inside VPNs are studied in [268].

5.3.2.5 Alternative Absolute Location Sources

Many approaches for relative positioning of nodes within wireless networks have been proposed. These make use of the wireless data transmission itself or use extra sensors, e.g., radar or lidar sensors [8, 331]. In contrast, only a limited amount of approaches to provide absolute location information, apart from GNSS based systems, at nodes is available.

Absolute positioning based on (public) WiFi networks is popular, especially for mobile phones. However, security of this kind of approaches is low, as shown, e.g., in [300]. Moreover,

coverage from such kinds of networks in many urban and highway scenarios is highly questionable. Thus, WiFi based position estimation seems infeasible to overcome the described GNSS security problem in VANETs.

Localization of a node within a mobile phone network is a well studied subject [196, 337]. However, its availability and accuracy are clearly bound to the coverage of mobile phone networks. Hence, the intended independence of an ad-hoc network from mobile phone network infrastructure would be called into question when presence of such kind of networks is required to obtain location information.

A general problem in regard to security related localization in VANETs is the lack of a commonly understood definition on how accurate location restrictions have to be checked. For numerous VANET based ADAS location requirements have been defined. In contrast, a corresponding requirement for the security functionality is still to be defined, and countermeasures to GNSS spoofing must be adapted to this requirement.

5.3.2.6 Short Lived One-Time PSCs

As outlined in Section 5.3.1.1, misuse of nodes for the generation of messages with future time stamps is possible. This is enabled by the availability of PSCs, which are (still) valid at a point in time being significantly in the future. Limiting the lifetime of each PSC would be a first step to resolve the issue.

One should note that the PSC management suggested for WAVE in [154] already implements this kind of short lived PSCs. In doing so, a validity time in the area of about ten minutes is proposed for each PSC. Moreover, nodes do not store multiple PSCs with an overlap in lifetime. This clearly avoids the possibility of a Sybil attack found in Section 5.3.1.1, as the nodes do not need to store multiple PSCs being valid during the same time span. Hence, the attack is avoided by system design.

However, the approach from [154] cannot avoid the generation of messages with future time stamps. To additionally counter this kind of attack, one also has to make sure that OBUs do not hold PSCs for future usage except for a quite short time span from the current point in time on. Thus, AAs need to have access to a secure time base (probably not GNSS time based) and may not issue PSCs for validity times, which are more than a short well defined time span in the future. Attacked nodes may request such PSCs with a validity time span in the future, as they regard their stored PSCs as being outdated.

To determine the time span for in advance generation of PSCs, one has to consider a trade-off between limiting the impact of the attack described above and the reliability of delivering new PSCs to using nodes before the ones stored in OBUs reach the end of their lifetime. This is required to ensure uninterrupted operation of the VANET communication system. Ideally, a fresh PSC is just delivered before its predecessor runs out of lifetime and gets used immediately. This would limit $t_{f,max}$ to the usage period of a PSC, i.e., to its lifetime (assuming that lifetime starts at the time of delivery to the OBU).

Thus, we propose to include the approach of short lived PSCs known from WAVE into the ETSI ITS system. Moreover, an extension to the PSC management scheme to ensure that the in advance storage period of PSCs is as small as possible should be included into WAVE as well as ETSI ITS.

The mechanisms described in Section 5.3.2.4 try to prevent the attacks from Sections 5.3.1.1 and 5.3.1.2 by avoiding the time manipulation. In contrast, the limitation of PSCs' lifetimes significantly limits the attacks' impacts, but cannot avoid the attack completely. An evaluation of the attacks impact on a real world ETSI ITS OBU is provided in the next section.

5.3.2.7 Secured Beacon Messages

Beacon messages are used to initiate the exchange of CAMs between nodes. These message are not signed [122]. Hence, an attacker can trivially generate them to cause a valid node being attacked by GNSS spoofing to generate CAMs. To increase the effort required to perform a successful attack, securing of beacons with the security profile used for CAMs [125] is proposed. This approach is based on the fact, that the pure reception of a message (beacon or CAM) is not enough to trigger the generation of a CAM at the receiver. Instead, the received message must arrive at the facility layer's CABS. Hence, input verification of received messages within the security entity on the network layer level can avoid the scenario of a bogus, i.e., not properly secured, beacon message triggering CAM generation at the receiver.

This approach can only be used to avoid attacks manipulating time into the future. For past time stamps, an attacker can just record valid messages and use them in a replay attack to cause a node to detect neighbors causing it to generate CAMs. However, this kind of replay attack is not possible for future time stamps. Hence, an attacker needs to successfully manipulate the internal time of at least two nodes, which will mutually cause CAM generation by sending beacons holding the manipulated time stamp. Hence, the minimum number of nodes, which an attacker needs to target with his GNSS spoofing attack is increased from just one (without secured beacons) to two (with secured beacons). However, once an attacker has obtained well signed messages with future time stamps, he can use these messages to cause CAM generation by other attacked nodes.

Securing beacon messages with a standardized ETSI ITS security envelope is considered unproblematic from a channel utilization point of view. Beacons are only sent in case no other VANET node is detected within communication range [122]. Hence, the wireless channel is unused, except for the transmitted beacon messages. Thus, increasing the size of a beacon messages by an added security envelope should not cause any problems in regard to channel load. Hence, the usage of secured beacons is recommended.

5.3.3 Experimental Evaluation

To evaluate the practical impact of the attacks described in Section 5.3.1 real world experiments with up to date OBU hardware have been performed. The used experimental setup is explained in Section 5.3.3.1 and obtained results are given in Section 5.3.3.2.

5.3.3.1 Test System Setup

To test the feasibility of the attacks outlined in Sections 5.3.1 the test setup described in the following is used. The OBU hardware Cohda Mk5 [68] is used as the device under attack. It runs the ETSI ITS conforming VANET protocol stack from the ezCar2X framework. The

attacker uses a wireless connection to deliver a malicious GPS signal, which is transmitted by a custom GPS generation and replay unit based on the Universal Software Radio Peripheral (USRP) platform [152]. A second OBU with always correct GPS input is used to send valid CAMs to the attacked OBU.

The following test cases are executed to resemble the identified attack surfaces.

1. A GPS signal holding a faked time stamp is provided to the OBU from the begin of its operation on.

   (a) The OBU holds PSCs being valid at received (faked) time. It is tested whether the OBU generates and emits messages with a time stamp equal to the faked GPS time stamps. Generation of messages with both past and future time stamps is tested.

   (b) The OBU has no access to a PSC, which is valid at the received GPS time. It is tested whether the OBU generates any message. This resembles a part of the DOS attack described above.

   (c) One GPS signal is provided to the OBU multiple times, and resets of the OBU are conducted before the GPS signal is provided anew. It is tested whether the OBU generates multiple sets of CAMs signed with different PSCs for the same future time span contained in the used GPS signal. This procedure enables an advanced attacker to perform a Sybil attack.

2. The unmodified GPS signal is provided to the OBU after its start-up. It is replaced with a GPS signal containing a time stamp significantly higher than the one in the first GPS signal, i.e., for the OBU this signal looks like coming from the future. Tests cases 1a and 1b are run. Furthermore, CAMs with a correct time stamp are sent to the OBU and reception of these valid messages is tested at the facility layer.

3. The test from 2 is ran, but after five minutes of receiving the manipulated time stamp, again the correct GPS signal is provided to the OBU. It is tested whether the OBU starts to send messages with correct time stamps again, after it receives the valid GPS signal again. This resembles part of the DOS attack from Section 5.3.1 and evaluates whether time stamp jumps in any direction are accepted by the OBU.

The results of these test cases are given in Section 5.3.3.2.

5.3.3.2 Test Results

An overview of the obtained results for test cases from Section 5.3.3.1 are provided in Table 5.1.

Results summarized in Table 5.1 show that all attacks providing a manipulated time stamp to the OBU lead to significant security problems. However, the attacker is not able to force a real time stamp jump after the device had already obtained a first GPS fix, i.e., after initial time synchronization was performed. Unfortunately, manipulation of this first time synchronization was always possible.

test case	observed result	security problem
1a	CAMs with manipulated time stamp generated	reliability and non-repudiation violated
1b	no CAMs generated	DOS weakness
1c	CAMs generated with different PSCs for same future time interval	Sybil attack via replay attack
2	CAMs generated with fast increasing time stamps until OBU's internal time is equal to provided GPS time	reliability and non-repudiation violated
	other CAMs accepted until time difference exceeds threshold, afterwards all received CAMs dropped	DOS weakness
3	at first like for 2	see above
	CAMs generated with slowly increasing time stamps until OBU's internal time is correct (again)	reliability and non-repudiation violated
	received CAMs accepted once time difference supersedes threshold, but all received CAMs dropped before	DOS attack successful until OBU's time is correct (again)

Table 5.1: Overview about test case results.

After the attacker transmits a GPS signal with a time stamp significantly far in the future, compared to the OBU's current system time (difference greater than ten minutes), the internal system time increased significantly faster (by a factor of more than two) than during normal operation. Thereby, the OBU's internal time synchronization mechanism tries to overcome the difference between system time and provided reference time from the spoofed GPS signal.

To analyze the cause of this behavior a custom time logger was run on the OBU during experiments. It was found that the system always starts up with its internal time being equal to the start of Unix time. This is followed by exactly one time stamp jump, which causes the system time to be equal to the time stamp contained in the first obtained GPS fix.

Further system analysis shows that the Cohda Mk5 uses gpsd to handle GPS data from the on-board NEO M8 GPS sensor [306]. Moreover, the ntpd alternative chrony runs on the system to provide time synchronization to GPS time. Furthermore, initial time synchronization is done with the help of a custom start-up script, which listens to the gpsd output (via the gpspipe tool) and performs a hard reset of the system time to the time stamp of the first obtained GPS fix. Afterwards, this script terminates and further time synchronization is left to the combination of chrony and gpsd. Thus, this findings clearly corroborate our experimental findings given before.

One should note that the described security issues are not caused by the used ETSI ITS implementation. Instead, they show a design problem of the current security architecture of VANET approaches, e.g., affecting both ETSI ITS and WAVE.

5.4 Limits of Privacy Caused by Protocol Data Sets

Privacy of nodes is an important requirement of future VANETs. This holds especially for vehicles, as this feature is only of minor interest for static RSUs. Privacy of participants is commonly achieved by using rapidly changing PSCs, as outlined in Section 2.2.3.

Attacks on privacy of nodes commonly try to perform tracking based on predicted node movement. However, this assumes that data sets within all protocol layers cannot be used to re-identify a vehicle after it performed a pseudonym change. The only suggested method proposed in prior work to guarantee this requirement is to change all unique identifiers on all protocol layers during a pseudonym change.

Data sets being constant before and after a pseudonym change and which are additionally different between nodes can undermine the effect of a pseudonym change. They can be used to create a (highly) characteristic fingerprint of a node. This fingerprint can be used to re-identify a node after a pseudonym change. Data sets which can be used for fingerprinting are called *characteristic constant data* in the following. Nodes with different fingerprints cannot belong to the same anonymity set. Hence, presence of characteristic constant data limits the level of privacy of nodes.

Section 5.4.1 provides an analysis of ETSI ITS in regard to the presence and impact of characteristic constant data on the various protocol layers. A comparison to the WAVE system is given in Section 5.4.2. Finally, Section 5.4.3 provides an evaluation on the impact of present characteristic constant data on the privacy gained by pseudonym changes, and suggested improvements of the state of the art. Parts of the topics discussed in this section are covered by prior work of the author published in [34, 35][4].

5.4.1 Protocol Analysis for ETSI ITS and Countermeasures

We divide our analysis on the influences on privacy caused by different data sets into the dedicated protocol layers. The facility layer is looked at first. Afterwards, the security envelope, as composed on the network layer, is studied. Finally, remaining protocol layers are considered.

5.4.1.1 Privacy Influence of Facility Layer Data Sets

Regular beacon messages within ETSI ITS are assembled in the CAM data structure. It is composed of deeply nested data structures with up to five hierarchy levels [119].

The following data sets are present within a CAM at the facility layer level. These are mandatory, if not explicitly stated otherwise. Presentation follows the hierarchical structure within the CAM's definition. A CAM contains data fields called

- protocol version,
- message ID,
- station ID,

[4]Contribution of the co-author is mainly related to collection of the vehicle manufacturers information about vehicles' sizes. The main contribution is from the author of this work.

- generation time,
- basic container, which holds
  - station type, and
  - reference position.
- A high frequency container is present in every CAM. It contains
  - dimensions (length and width of vehicle),
  - vehicle's dynamics,
  - optional data including
    - more dynamics information: steering wheel angle, lateral acceleration and vertical acceleration,
    - acceleration control,
    - lane position, and
    - performance class.
- A low frequency container (optional) is sporadically included by all nodes and holds
  - vehicle role,
  - exterior lights status, and
  - path history.
- Usage of a special vehicle container is optional. Presence of such an container enables to distinguish its sender from other nodes with an ordinary vehicle role in the VANET. Each special vehicle type uses its own kind of container. It is sporadically included in CAMs. Inclusion only happens if the low frequency container is not included, i.e., there is at most one optional container in a CAM.

The message ID is a fixed value for CAMs, and the station ID is changed during the pseudonym change procedure. Thus, these data sets are not regarded in the following, as they do not offer an unintended possibility for continuous node tracking before and after a pseudonym change.

Data sets from within a CAM with possible influence on privacy of their sender are discussed in detail in the following. Furthermore, proposals for privacy enhanced usage of such data sets are given.

Protocol Version The protocol version is not changed during a pseudonym change, and can be assumed to be constant for all nodes at the beginning of deployment. However, it is clearly characteristic constant data. Moreover, over time multiple versions may be present within VANETs at the same time, differentiating nodes into distinguishable anonymity sets. Hence, the presence of different versions should be avoided, even in case mutual compatibility is maintained from a functional point of view..

Generation Time The generation time is different for each CAM from the same node. However, the temporal difference between two sequential beacon messages is defined by standards [117]. Neither ETSI ITS nor WAVE define any change to the transmission interval during the pseudonym change process.

A common assumption is that clocks of nodes within a VANET are well synchronized using GPS [325] (see also Section 2.5.2). Hence, time intervals between CAM generation within individual nodes should be quite stable. Moreover, within a set of nodes, the generation times of CAMs should be randomly distributed leading to an even distribution of used time stamps. Such time stamps are generated and transmitted twice within the protocol stack. Once with millisecond resolution within the facility layer CAM itself, and additionally with microsecond resolution within the security envelope [117, 125]. Hence, collisions in this data field, which could confuse an attacker, are quite unlikely. Therefore, an attacker can track nodes just based on their sequence of CAMs' generation time stamps with high probability.

In case of WAVE, the sending interval of BSMs is fixed. For CAMs, it is determined by multiple parameters and can be in the range from 1 to 10 Hz. However, the current interval is additionally contained in each CAM [117]. This allows an attacker to easily use this information to avoid being confused by the variable CAM transmission interval.

Furthermore, the time stamps are set above the MAC layer. Hence, tracking capabilities of the attacker are not limited by the probabilistic MAC layer's channel access mechanism. Only the actual sending time is somehow randomized by the probabilistic CSMA-CA scheme, which is used within ITS-G5 and IEEE 802.11p.

Two approaches to overcome the described vulnerability are proposed. Both require the nodes cooperating for the PSC change to use the same sending frequency before and after the change for a minimum time span, e.g., one second, at least. This avoids tracking based on an individual beacon generation interval of a dedicated node.

In the first approach, one reduces the accuracy of the generation time within the security envelope and the facility layer's CAM data structure to the commonly used transmission interval, which is 100 ms for BSMs, and 100 ms - 1 s for CAMs. This causes all CAMs sent within the same time interval to use equal time stamps. Thus, nodes cannot be differentiated based on CAMs' time stamps.

The security entity does not need to determine the sequence of received messages according to standards. Moreover, the validity time spans of PSCs are specified with full second resolution. Hence, there is no need to use a high precision time stamp with microsecond resolution for the generation time contained in the security envelope. It should be substituted by one with less accurate resolution. A side effect would be a possible reduction of the security envelope's size by four bytes [125].

However, the acceptable discretization of the time stamp at the facility layer is limited by application's requirements for accurate trajectory modelling of cooperating nodes. Hence, the following proposal avoids such further discretization.

For the second proposal, immediately after the pseudonym change the next transmission has to be delayed by a random time span. The length of this waiting time should be in the order of the normal time difference between two successive transmissions. For example, WAVE would use values between zero and 100 ms for BSMs. As a consequence, an attacker cannot calculate

the next message generation time stamp and gets confused. The impact on higher level layers, e.g., applications, should be low. From their perspective a maximum delay looks just like a single missed message from the other node.

Station Type The station type associates a node to some generic class, e.g., passenger car or light truck. This constant data set is clearly characteristic constant data. Thus, cooperative PSC switching strategies have to take it into regard, i.e., cooperation between nodes with different station types does not increase their privacy, as they belong to different anonymity sets.

Reference Position A node's current position measured at its reference point (see [108]) is available in each CAM. Prior work already showed that this information can be used to bypass simple PSC change strategies [146, 325]. Hence, the advanced PSC switching strategies, like the ones suggested in these references, should be used.

Vehicle Dimensions A node's length and width dimensions are included in each CAM with a resolution of 0.1 m [117]. These values stay constant at least during one journey of a node. Hence, they are characteristic constant data. The dimensions of a vehicle may change from one journey to another, e.g., by extension with a trailer.

Privacy aspects of disseminating a node's dimensions have hardly been regarded in prior work, except of some remarks in [274]. They can be studied by looking at the characteristics of currently sold vehicle models in Germany. The individual models can be assigned to the discretization steps of length and width, which are defined in [117]. Required data is publicly available, e.g., from the German Kraftfahrt-Bundesamt [198], which holds information on the share of different vehicle types separated into OEMs and their models. Additionally, public information from the 45 different OEMs present in [198] was used to obtain the individual dimensions of vehicle models.

These data sets characterize the overall traffic in Germany caused by newly sold cars. Foreign cars traveling on German roads are excluded from this data set. However, it should still provide a reasonable estimate about the overall distribution of vehicle's dimensions.

The conduced analysis finds that 73% of all vehicle models share a common combination of width and length with at least one other model. These vehicles represent a market share of 75%. Hence, for a share of 25% one can determine the model directly from its given discretized dimensions. Even the most common set of vehicles, with length 4.3 m and width 2.0 m, represents only 17% of all cars. Therefore, most nodes in a VANET can be (re-)identified within their close vicinity based on their transmitted dimensions, which helps an attacker to track nodes.

Dissemination of node dimensions clearly decreases the probability to find proper (i.e., indistinguishable) partners for a cooperative pseudonym change. Further discretization of sent dimensions to, e.g., 0.3 m would significantly improve the situation for many nodes (see also Section 5.4.3), but cannot help vehicles with outstanding dimensions.

Vehicle Dynamics The data sets of longitudinal acceleration, curvature (consists of curvature value and confidence), curvature calculation mode and yaw rate are present in each CAM. They model a node's dynamic behavior. The curvature calculation mode is a value, which is unlikely

to change for an individual node, and may differ between nodes. Hence, it should be regarded as characteristic constant data.

The remaining values model a node's trajectory. Many approaches for modeling and predicting vehicular trajectories exists, e.g., [10, 164]. In case of pure tracking, i.e., no realtime interaction between attacker and nodes, the attacker does not need to process the received data in real time. Hence, he can apply computationally expensive, but accurate and complex, movement models. As shown above, an attacker can determine either the vehicle's type directly or a group of possible vehicle types. This information can be used to tune the parameters of the movement model causing it be very accurate. Moreover, movement prediction must only work well for a short time span as the CAM generation rate is at most one second.

To evaluate the impact of using an advanced movement model on the attacker's tracking ability one should prefer to use data obtained from real test drives, instead of pure simulator output. This is because simulators like the well known SUMO use a predefined movement model for nodes. Thus, tracking these simulated nodes with a movement model, which fits the one used to generate their movement, will probably yield unrealistically high success rates.

Optional Data within the High Frequency Container Six data sets may be optionally present in a CAM's high frequency container. Three out of them (steering wheel angle, lateral, vertical acceleration) can be used to improve the movement model described above.

The remaining three values (acceleration control, lane position, performance class) each describe a node's individual feature. These can be expected to change quite slowly, i.e., they should be regarded as characteristic constant data.

All the optional data sets can be added or removed individually. Hence, also the combination of present data sets may differ between nodes. Therefore, usage of each extra value will increase the change that a particular node uses a unique set of data sets inside its current vicinity. Thereby, it will strip itself from finding proper partners for performing a privacy preserving pseudonym change. Hence, the information about presence of optional data fields has to be considered characteristic constant data.

Optional Containers In addition to the basic and high frequency container, the low frequency container is distributed cyclically, but not in every single CAM. It contains the vehicle role, exterior lights and path history fields. Detailed inclusion rules can be found in [119].

Typically, the status of exterior lights changes slowly. Thus, this data set is characteristic constant data. However, many nodes can be expected to share the same value.

The path history field should obviously be erased when a pseudonym change occurs or the inclusion rate of the container has to be such low that sequentially sent values of this field cannot be linked. Otherwise the attacker can simply link pseudonyms based on this data. However, the current standards do not specify such behavior, but it is recommended in [60].

In case of an uncommon vehicle role, e.g., rescue vehicle, the corresponding extra container is cyclically included in the CAM. Density of such vehicles in ordinary traffic is usually low. Therefore, an attacker can easily track them just based on the presence of their dedicated containers within their CAMs.

5.4.1.2 Privacy Influence of Security Envelope Data Sets

In case the security profile for CAMs is used, the following data sets are present within the security envelope [125]. Thus, they may influence the privacy of the sender. Mandatory data sets include,

- protocol version,
- signer info, which holds exactly one of the following elements:
  - HashedId8, i.e., the hash value of a certificate,
  - certificate, usually a PSC., or
  - certificate chain, which is composed of an array of at least two certificates.
- generation time,
- ITS-AID: identical for all CAMs, thus one cannot differentiate nodes based on its content.
- certificate request list (optional), and
- signature.

All data sets except the last one are stored in the header part of the security envelope. Only the signature is stored in the trailer part. Moreover, the signature is currently the only data structure which may be stored in the trailer [125].

Many more data structures are defined for usage within the security envelope [125]. However, they are not used by any currently defined security profile. Thus, they are not regarded in this analysis.

The data sets with (possible) influence on privacy of nodes are discussed in detail in the following. Moreover, suggestions for privacy enhanced usage of these data fields are provided.

Protocol Version The used protocol version can be assumed to be constant for all node at the beginning of deployment. However, over time multiple versions may be present within VANETs at the same time. This value is constant for an individual node over a long time, and is especially not changed during a pseudonym change. Hence, it is clearly characteristic constant data. Thus, the presence of different versions should be avoided, even in case mutual compatibility is maintained. This finding is equal to the one for the protocol version information, which is contained with the CAM data structure, given above.

Certificate The certificate data structure is clearly the most complex sub-structure of the security envelope. It contains the following data sets.

- version, with a privacy impact equal to the one of the protocol version discussed before,
- signer info, which can either be

  - empty in case the certificate is a root certificate, which does not need to be signed by any other certificate. However, root certificates are not distributed by inclusion in CAMs.
  - a HashedId8 for all non-root certificates. A node can be assumed to be assigned to a dedicated AA to receive its PSCs. Thus, the hash value of the AAC from this AA is present in this data field. It is characteristic constant data, as discussed in more detail later on.

- validity restriction(s), which are discussed in-detail in the following,
- subject attribute(s): This data field holds subject type and public key of the certificate. This key is randomly generated and the subject type is fixed for all PSCs. Thus, there is no possibility to link PSCs, and thereby pseudonyms, based on the subject attributes.
- subject info: This data field holds a fixed value for all PSCs. Hence, it provides no possibility to track nodes.
- signature: Similar to the signature of the whole security envelope, the content of this field does not add any new tracking related information, as the signer of the certificate is already identified by the signer info field mentioned above.

The privacy impact of the data fields given above is discussed in more detail in the following.

Signer Info of a PSC A PSC's signer info field identifies its issuing AA. Syntactically, this can be either done by the hash digest of the AAC or by the full AAC. Both uniquely identify the AA. Within ETSI ITS, PSCs may only hold the AAC's ID instead of the full certificate to keep PSCs short [125].

ETSI ITS and WAVE allow for a multitude of CAs (or AAs) to exist. In practice, such authorities will be probably operated by the car manufacturers, i.e., one CA per OEM. Unfortunately, this leads to a privacy problem. The signer information is caused to be characteristic constant data, as a vehicle's manufacturer does never change. An attacker can directly obtain a node's OEM from its PSC and use this information to distinguish nodes. The probability of a node to use PSCs from different AAs is very low. In contrast, it is very likely for a node to use only PSCs issued by the same AA. Clearly, nodes manufactured by low volume OEMs are particularly vulnerable to tracking based on their manufacturer's identity.

There are mainly two countermeasures, to limit the usability of an AA's identity for an attacker. At first, one could increase the overall number of AACs, by using a multitude of them for each single AA. Thereby, the effort for an attacker to keep track of all AACs and their mapping to OEMs would increase. However, this would significantly increase the effort for AAC distribution to all nodes (see also Section 4.2.2) for a small security gain.

Secondly, in a converse approach, one could limit the number of AAs. An ideal choice would be to have only a single AA using a single AAC. Thereby, one would clearly remove the privacy problem described above, as attackers cannot distinguish nodes based on their AA anymore. To realize this approach, OEMs have to cooperate and use a common AA. As European OEMs plan to establish a common root CA for Europe, this seems to be a feasible approach. To limit the

number of PSCs signed by a single AAC, one could significantly limit its lifetime, e.g., to some weeks. A new AAC could be deployed during PSC refills.

Moreover, one should coordinate the lifetime of a common AAC with the lifetime of its issued PSCs. Thereby, any possibility to distinguish PSCs based on their issuing AA should be ruled out. Furthermore, the number of AACs to be stored securely inside each node is kept (very) low. This helps to limit costs of required HSMs inside OBUs.

The privacy gain from using only a single AAC is evaluated in Section 5.4.3.

Validity Restriction(s) The only mandatory validity restriction of a PSC is a limited validity period. Such a validity period is defined by start and end of life time stamps. For both values an accuracy of one second is used. According to the PSC update (or refill) scheme from ETSI ITS and WAVE, PSCs are delivered from a CA (resp. an AA in ETSI ITS) to a node upon its request [102,176]. Many remaining details are implementation specific, as these are not covered by standards. However, there exists a possible pitfall for privacy of nodes, which is caused by the mentioned time stamps.

This PSC usage privacy issue arises from the planned way of (re-)using PSCs in Europe. Following this approach, each vehicle uses a pool of PSCs, which are (re-)used until the full pool gets updated [24,303]. The update period will probably be in the order of months. Reusing of PSC has received criticism [251], but the following newly discovered weakness affects PSC pools even in case no re-usage of PSCs takes place.

A one time usage approach for PSCs is described in [154] for WAVE. Following this approach, each PSC is only used once and its validity period is the order of minutes. However, this approach introduces significant overhead in the ITS system for PSC distribution. Either vehicles require frequent connections to the backbone CA or a huge buffer filled with PSCs for future usage. Even doubling the proposed validity period of five minutes [154], this would still require a maximum amount of 144 PSCs per day. To protect the locally buffered PSCs, these have to be stored in secure memory, e.g., inside an HSM. However, adding more memory to an HSM increases its price. Moreover, many issued PSCs will stay unused, as their lifetime elapses, while the node stays unused. One would have to know usage times of each vehicle in advance to avoid that, which is hardly practicable. In case PSCs are updated in whole sets anyhow, the following approach for securing re-usage of PSCs is recommended.

PSCs are generated inside an CA upon request of a node. A straight forward implementation would take the same time stamp, e.g., the current time at the CA when the request arrives, and use it as the common start validity time stamp of all signed PSCs for one node. However, this means that all PSCs of a set delivered to a node have a very similar (or even the same) start validity time stamp. Thereby, this information is caused to act as characteristic constant data. Moreover, this time stamp will be different for most cars with a very high probability, as there is no timed synchronization between PSC refill requests. Hence, this time stamp could even serve as a unique identifier for the node. The same holds for the end of validity time stamp.

Nodes using obtained PSCs have no possibility to protect themselves against an attacker using validity time stamps for tracking them. They cannot change the content of a PSC on their own, without invalidating its signature. Thus, countermeasures have to be taken within CAs.

A straight forward solution would be to apply discretization for time stamps defining the

validity period of PSCs. In doing so, globally synchronized time slots for validity periods could be used. Thereby, all nodes receive a set of PSCs with a common validity period. Consequently, all nodes meeting on the street use common values in these data fields, which removes the possibility to distinguish them. Only basic time synchronization between CAs is required to realize this approach.

Generation Time The generation time is present in both the security envelope as well as the facility layer CAM content. However, the time stamp's resolution is much higher within the security envelope. Microsecond resolution is used there, while only millisecond accuracy is used within the CAM. For an analysis of the privacy impact of this data field see the corresponding discussion in Section 5.4.1.1 for the generation time stamp inside the CAM data structure.

Certificate Request List A node requests up to six unknown certificates (PSCs or AACs) by using the last three bytes of their hash values. Standards are unclear about when to remove entries from the request list [125, 176]. An algorithm for maintenance of this list is suggested in Section 4.2.1. It proposes repetition of requests, i.e., entries in the list can be present in sequential messages. It can be expected that the current set of certificates requested by a node is highly discriminative between nodes. Hence, the certificate request list should be flushed during a pseudonym change.

Signature The signature field within the security envelope's trailer holds metadata for interpreting the digital signature and signature itself. Most parts this data structure are fixed and signatures of multiple messages can only be linked together using the respective public key, which is changed during the pseudonym change procedure. Hence, a digital signature does not carry any extra privacy related information, in comparison to the public key hold in the corresponding PSC.

However, the encoding type of the used ECC point may vary in general, but can be assumed to be constant for a particular vehicle. Hence, this encoding type is characteristic constant data. There are four options for the ECC point type within ETSI ITS. These relate to three different options of representing an ECC point, by either given only its x-coordinate, or the x-coordinate together with the sign of the y-coordinate, or both coordinates. The core difference between options is enabled or disabled ECC point compression. In the worst case, with three vehicles cooperating during a PSC change, and all of them using a different ECC point type, this information is already enough to render the pseudonym change useless. Thus, the standard should only allow only one option to be used instead of four. This restriction is also suggested in [237], to ease parsing of the security envelope.

Summary Overall, many occurrences of characteristic constant data within the security envelope of ETSI ITS have been found. Each single finding reduces chances of nodes to find proper cooperation partners for a privacy conserving PSC change. Hence, the proposed approaches to remove characteristic constant data from the security envelope should be applied.

5.4.1.3 Privacy Influence of MAC/PHY, Network and Transport Layer Data Sets

The node identifiers at the PHY/MAC and network layers depend on the pseudonym, as outlined in Section 2.2.3. Thus, presence of these identifiers does not add extra information, which could be used for node tracking.

Network layer meta data holds some data sets, which can also be found within a CAM or BSM. This includes position information alongside with speed, heading and a time stamp [119]. However, no extra information is obtained by the attacker using this duplicated data sets.

The transport layer uses only a port ID, which is equal for all message from the same type, e.g., CAMs or BSMs, at all nodes. Thus, this data set cannot be used to differentiate nodes.

5.4.2 Comparison of Influence on ETSI ITS and WAVE

The security functionality and used data sets are very similar in ETSI ITS and WAVE [125, 176]. Thus, the impact on privacy of these data sets within WAVE is the same as outlined in Section 5.4.1 for ETSI ITS.

Network and transport layer for safety critical VANET communication within WAVE, e.g., for BSMs, provide much less features in comparison to ETSI ITS. Thus, the amount of present meta data for these layers is very low. The present data sets do not provide an attacker with extra information, not already known from the security envelope's content.

ITS-G5 used for ETSI ITS shares the same sets of meta data with IEEE 802.11p, which is used together with WAVE. Hence, privacy impact of PHY/MAC layers in WAVE is very limited, as described in Section 5.4.1 for the case of ETSI ITS.

The main difference between ETSI ITS and WAVE exists on the application (or facility) layer. The structure of data sets present with a BSM is significantly different from a CAM. However, the set of mandatory data is quite similar [17, 119]. Especially, the identified characteristic constant data sets exist in both message types, e.g., vehicle dimensions.

5.4.3 Evaluation of Privacy Loss and Countermeasures

To evaluate the actual impact on privacy of nodes the simulation environment described in Section 3.3 is used. Moreover, characteristic constant data from the data sets of

- AA issuing a node's PSCs with the assumption of one AA per OEM (see, e.g., [264]) and
- vehicle dimensions (length and width) are considered.

Section 5.4.3.1 describes the metric used for evaluating the impact of present characteristic constant data on node privacy, especially in regard to node tracking instead of a conducted pseudonym change. Regarded privacy change strategies are given in Section 5.4.3.2. Obtained results from the conduced evaluation are discussed in Section 5.4.3.3.

5.4.3.1 Vehicle Uniqueness as a Privacy Metric

Common pseudonym change schemes are based on the assumption that broadcast data cannot be mapped to an individual vehicle except of the changed identifiers. However, results from Sections 5.4.1 and 5.4.2 show that presence of characteristic constant data violates this assumption.

To evaluate the impact of this findings on privacy of VANET nodes, a the metric called *vehicle uniqueness (VU)* is introduced. It measures the difference between the fingerprints of nodes from their vehicular environment in regard to data observable by an attacker.

Prior work shows lower tracking success rates with rising traffic density and longer distances traveled during a cooperative pseudonym switching maneuver [303]. However, one can expect that this only holds in case an attacker has no extra information for re-identification of a node after a pseudonym change. VU is a metric for availability of such extra information. In case a vehicle is unique inside the area of pseudonym changing, i.e., the cardinality of its anonymity set is equal to one, the attacker is always able to track it. This is independent of the used pseudonym change strategy.

To calculate VU, a so called *exposed feature vector* $\boldsymbol{e}_i$ is defined. It holds all available characteristic constant data, which is available for each vehicle. $i \in \boldsymbol{I}$ relates to a dedicated node within a group of nodes $\boldsymbol{I}$ ($|\boldsymbol{I}| \geq 1$) cooperating during a pseudonym change. VU is defined by

$$VU = \mathbf{Pr}\{|\{x|\boldsymbol{e}_x = \boldsymbol{e}_y; x \neq y; x, y \in \boldsymbol{I}\}| = 0\}\,. \tag{5.1}$$

Thus, $VU \in [0; 1]$ is the probability that there is just one node within $\boldsymbol{I}$ showing one dedicated $\boldsymbol{e}_i$ to the attacker. Nodes sharing exactly the same $\boldsymbol{e}_i$ are indistinguishable for an attacker regarding characteristic constant data, or in other words such nodes form an anonymity set. Thus, these vehicles are proper candidates for cooperation during a pseudonym change.

5.4.3.2 Regarded Privacy Change Approaches

For the conducted evaluation three different pseudonym switching schemes are taken into regard. These are

1. uncoordinated pseudonym switching (ETSI ITS and WAVE) with $|\boldsymbol{I}| = 1$ with high probability,
2. mix zones with $|\boldsymbol{I}|$ depending on traffic flow and size of the mix zone and
3. pure silent periods with $|\boldsymbol{I}|$ depending on traffic flow and length of silent periods.

Within current standards pseudonym changes are uncoordinated, as every node decides on its own when to perform the change without including information from other nodes in its decision process. Hence, the probability that the trajectories of two nodes intersect while both change their pseudonyms between two successive beacon emissions, e.g., a time span of 100 ms in WAVE, just by chance can be expected to be very low. However, this would be required to confuse an attacker who tracks the nodes' movement. Absence of such behavior leads to $|I| = 1$, i.e., there is no simultaneous change of pseudonyms by nodes within close vicinity.

Results of an evaluation using the proposed VU metric as well as the concept of anonymity sets are given in the following.

5.4.3.3 Evaluation Results

In the following three different system parametrizations are considered. Firstly, a system using a multitude of AAs is studied. This resembles the currently planed way of VANET deployment

based on ETSI ITS and WAVE. Secondly, the approach for usage of a single AA, as proposed above, is studied to show its significant improvement potential on privacy of nodes. Afterwards, privacy improvement by further discretization of vehicle dimensions is studied. Finally, a summary about overall results achieved in the given evaluation is provided.

Multiple Authorization Authorities In this section we include the following data sets into $\boldsymbol{e}_i$:

- AA of PSCs, with the assumption of one AA per OEM (see also Section 5.4.1.2), and
- vehicle dimensions (see also Section 5.4.1.1).

It is assumed that all vehicles manufactured by the same OEM use the same encoding of ECC points (see Section 5.4.1.2). Thus, this data does not influence VU in our case and is not taken into regard in the following. The remaining characteristic constant data sets from Sections 5.4.1.2 and 5.4.1.1 are assumed to be identical for all nodes. This leads to a best case assumption for privacy of nodes, i.e., a worst case assumption for the attacker. Moreover, we assume that the probability of two nodes within $\boldsymbol{I}$ sharing a common value of $\boldsymbol{e}_i$ ($|\boldsymbol{e}_i| = 3$) only depends on the share of their particular model within the set of all vehicles.

Vehicle distribution is taken from [198] to estimate VU. Furthermore, an analysis of node's dimensions for the models of different OEMs (see also Section 5.4.1.1) shows that the data included in $\boldsymbol{e}_i$ allows to uniquely identify the model of a vehicle, e.g., as VW Golf VII, from a single CAM with included PSC. Thus, one can calculate the probability to encounter a node with a particular $\boldsymbol{e}_i$ from the mentioned vehicle distribution data set. An illustration of the distribution of anonymity sets resulting from the considered $\boldsymbol{e}_i$ is given in Figure 5.7 using a Cumulated Distribution Function (CDF).

One can see from Figure 5.7 that the members of the majority of anonymity sets has only a low share on the overall number of nodes. Hence, those nodes can only expect a low level of experienced privacy, i.e., tracking of of them is easy.

The number of nodes encountered during a PSC switching maneuver $|\boldsymbol{I}|$ is varied by varying the traffic density (given in $\frac{vehicles}{kilometer}$) and size of mix zones or length of silent periods, respectively. The traffic density is varied from 16 to 45 $\frac{vehicles}{kilometer}$ per lane following [146] to represent both low volume traffic as well as a jammed setup. We use parameters from [303] for the size of mix zones (25 m - 400 m), length of silent periods (1.25 s - 20 s), and velocity range (0 - 250 $\frac{km}{h}$). This leads to a range for $|I|$ from 1 to 65 vehicles. Obtained results for VU are displayed in Figure 5.8.

The *best* case in Figure 5.8 relates to the most common vehicle model. It is the least unique one within the set of all vehicles. However, only about 7.7% of all vehicles can profit from the good results for this model having a high chance to find indistinguishable partners for a cooperative pseudonym change. In contrast, the *worst* case relates to the least common car. It has only a very low probability to find proper partners to perform a secure PSC change.

Figure 5.8 shows that the value of $1 - VU$ increases alongside with $|\boldsymbol{I}|$. However, for an *average* vehicle it is very low for all regarded values of $|\boldsymbol{I}|$. Moreover, combinations of high velocity and high traffic density, which lead to high values of $|\boldsymbol{I}|$, rarely occur in practice. Thus, VU will exceed 99.9% in most real world scenarios with moderate traffic densities and vehicles' velocity.

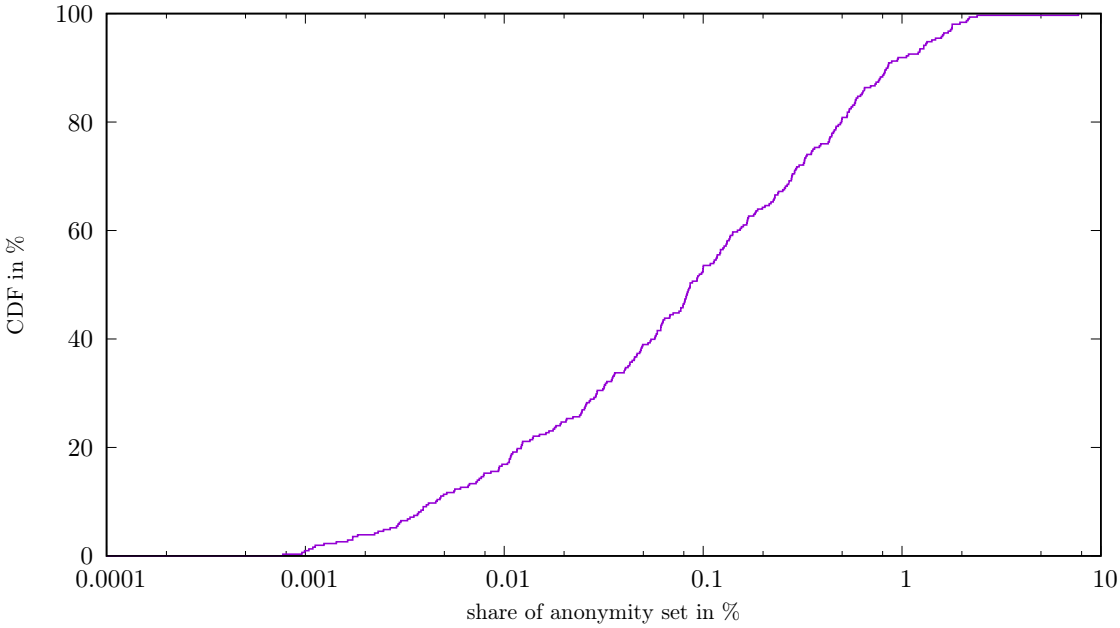


Figure 5.7: CDF of anonymity sets resulting from $|\boldsymbol{e}_i| = 3$ and standardized data sets.

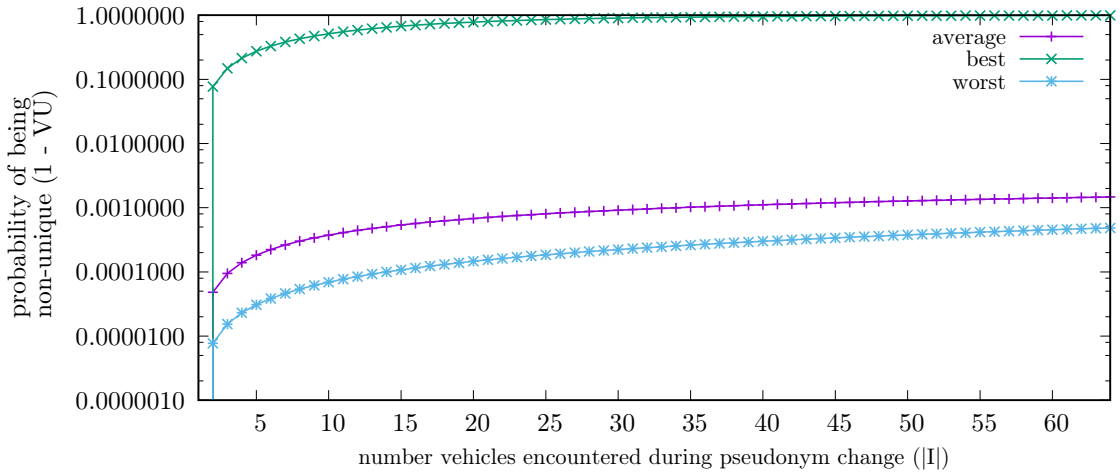


Figure 5.8: Vehicle uniqueness during pseudonym change with $|\boldsymbol{e}_i| = 3$.

This means that the attacker can trivially track an average node even after a performed PSC change with more than 99.9% probability, just based on characteristic constant data. In combination with other techniques from prior work, such as trajectory based tracking, hardly any privacy of nodes can be expected to remain.

Higher values of $|\boldsymbol{I}|$, than the ones used above, would relate to unrealistically dense traffic flow or extending the size of mix zones and length of silent periods to values rendering higher level applications unusable [303]. One should note that even medium size mix zones have been

shown to cause significant performance degradation of VANET based ADAS [205]. Calculation of VU is independent of the pseudonym switching strategy, but the achievable size of $|\boldsymbol{I}|$ differs. While cooperative PSC switching strategies can adjust it, uncoordinated ones cannot do so.

The obtained data on vehicle uniqueness shows that the presence of characteristic constant data is able to render PSC changes during driving almost useless. An attacker can almost always re-identify vehicles based on these data sets after the PSC change. To counter the identified tracking possibility, the approach of using a common AA for all vehicles is looked at in the next section.

Common Authorization Authority To reduce vehicle uniqueness the usage of a single AA for all nodes, as proposed in Section 5.4.1.2, is considered in the following. In contrast to the section before, the exposed feature vector $\boldsymbol{e}_i$ only holds the nodes' dimensions, i.e., $|\boldsymbol{e}_i| = 2$. The AA's identity is no longer present in e_i, as it is identical for all nodes.

The proposed privacy improvement via a commonly used AA, which leads to a common AAC, changes the anonymity sets of nodes. The CDF of anonymity sets corresponding to the $\boldsymbol{e}_i$ considered here is given in Figure 5.9.

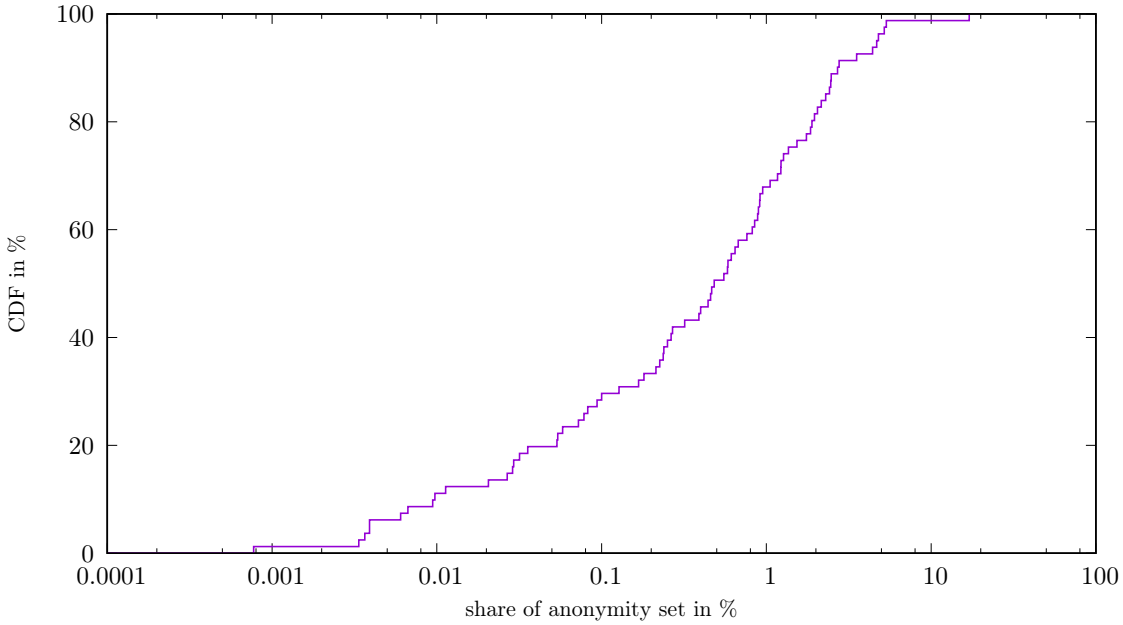


Figure 5.9: CDF of anonymity sets resulting from $|\boldsymbol{e}_i| = 2$ and standard vehicle dimension's accuracy.

A comparison of the results given in Figures 5.7 and 5.9 shows that the overall number of anonymity sets is reduced by the taken approach. Hence, the share of members of many anonymity sets on the overall amount of nodes is increased, i.e., some anonymity sets get joined to form a common anonymity set by the suggested approach. Hence, privacy of affected nodes is improved. However, there are also anonymity sets, which stay unchanged. Thus, the experienced level of privacy of their members is unchanged.

The distribution of global anonymity sets resulting from the chosen e_i is also illustrated in Figure 5.10 based on values from [198] and vehicle dimensions. Each anonymity set is labeled with the width / length value pair, which characterizes the corresponding set. One can see that the number of sets is high, and there are many sets with a (very) small share one the overall set of all nodes. Nodes from such small sets can be expected to hardly encounter other nodes from their own set while driving on streets. Hence, their level of privacy is small, as tracking of them is easy. However, there is a significant improvement in comparison to a system with a dedicated AA for each OEM, which yields much more anonymity sets with lower market shares.

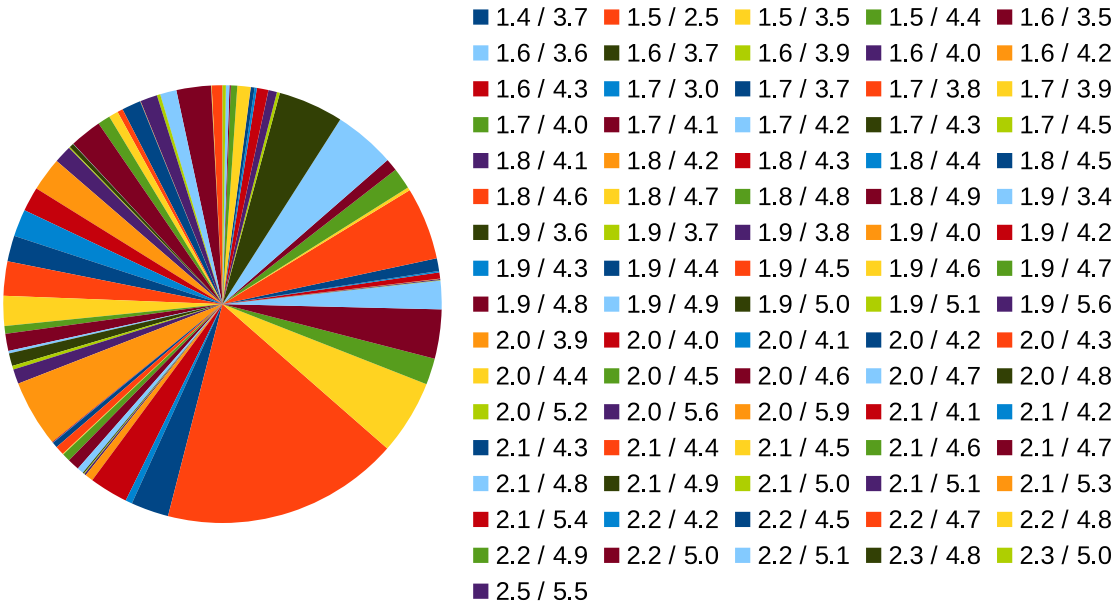


Figure 5.10: Anonymity sets of cars based on their exposed dimensions in CAMs.

The same vehicle distributions and traffic scenarios as before are used for evaluating the proposed privacy improvement technique. Thereby, well comparability of both approaches is ensured. Obtained results for VU in the system using $|\boldsymbol{e}_i| = 2$ are shown in Figure 5.11.

Comparison of the results from Figure 5.11 to the ones from Figure 5.8 show that using a single common AA reduces VU (increasing $1 - VU$) by a factor of about eight. Hence, privacy of nodes is significantly increased. Results for the least common vehicle model are unchanged. However, for the most common and average vehicle models an improvement of privacy is achieved, although uniqueness of an average node is still high. The most common group of indistinguishable nodes contains about 17.0% of all vehicles for this approach. These findings are in line with the ones in regard to the distribution of anonymity sets described above. To further boost the privacy of nodes, an approach changing discretization steps of nodes' dimensions is studied in the following.

Further Discretization of Vehicle Dimensions and Common Authorization Authority A further mechanism to reduce VU is to reduce the accuracy of nodes' dimensions included in

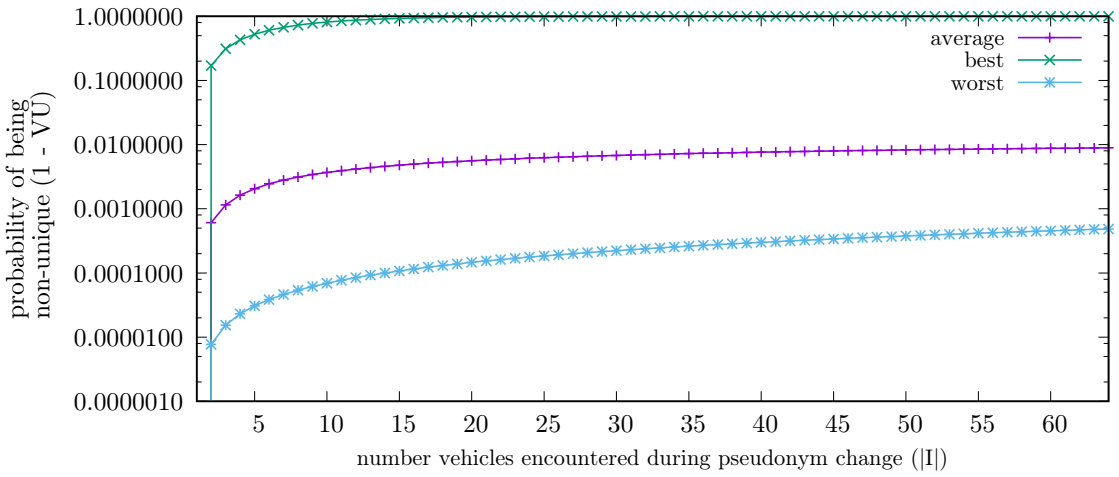


Figure 5.11: Vehicle uniqueness during pseudonym change with $|e_i| = 2$ and standardized vehicle dimensions' accuracy.

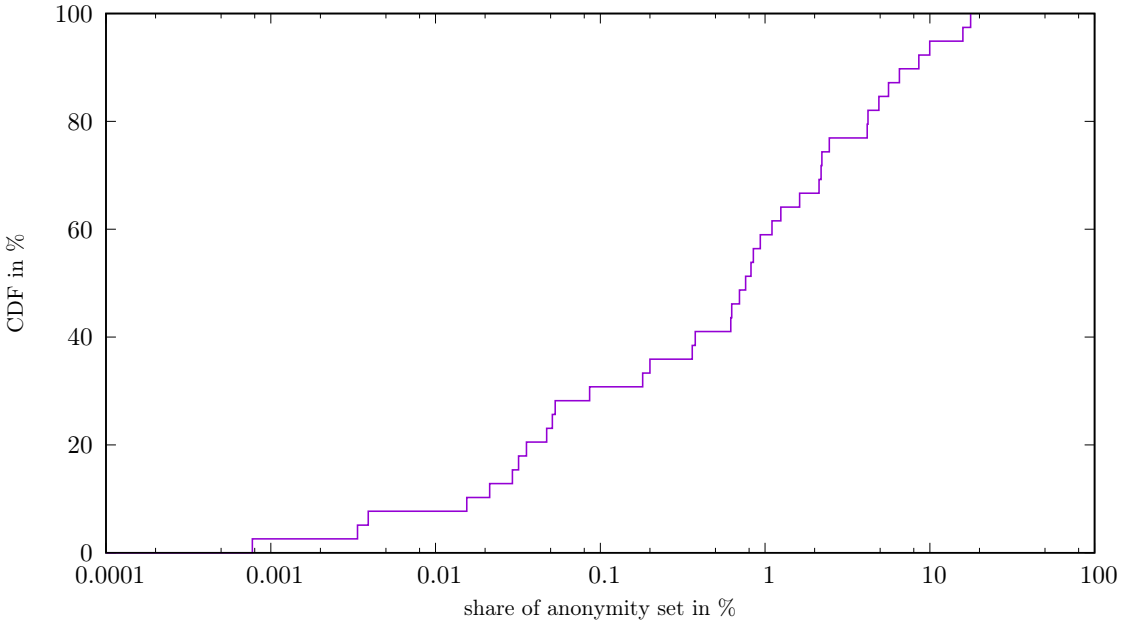
CAMs, as suggested in Section 5.4.1.1. To evaluate its impact the same setup as in the two sections before is used.

The resolution of node's dimensions length and width are further discretized to resolutions of 0.2 m and 0.5 m. With decreasing resolution the data quality available for applications is lowered. However, no detailed requirements regarding this parameter set have been published so far. Thus, future work is required to obtain a trade off between privacy and application requirements in regard to this dedicated data set.

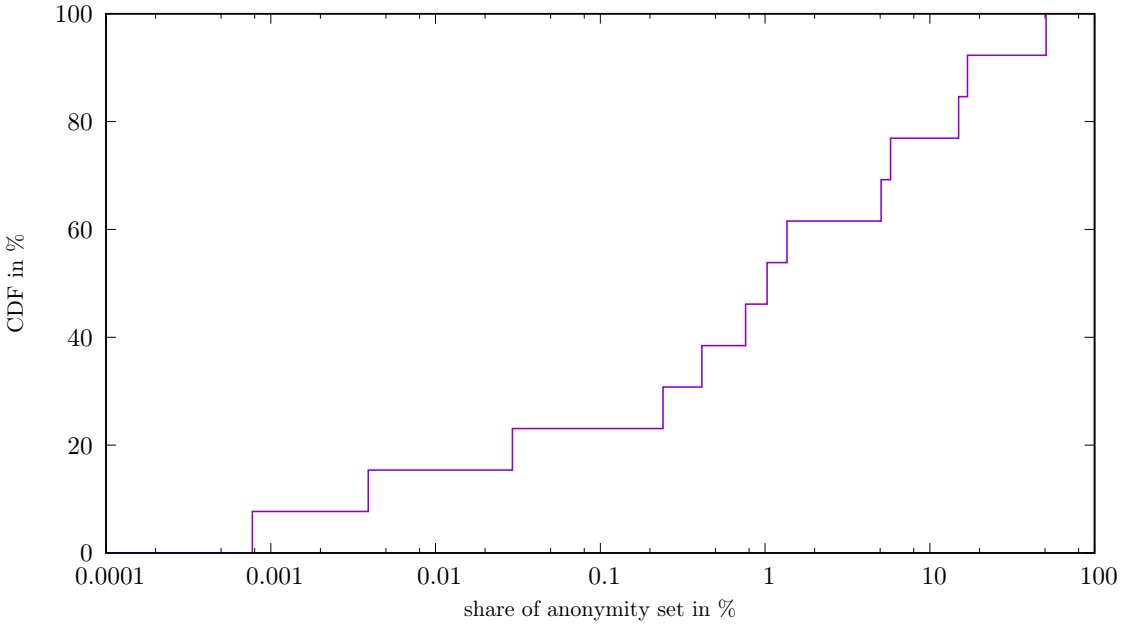
CDFs of anonymity sets resulting from lower resolutions of vehicle dimensions are given in Figure 5.12. Comparison of the results from Figure 5.12 with Figures 5.7 and 5.9 shows that the number of anonymity sets can be significant reduced by the suggested discretization approach of vehicle dimensions. Moreover, the share of anonymity sets' members on the overall share of nodes is significantly increased for almost all nodes. Hence, privacy of such nodes is improved.

The distributions of anonymity sets for dimensions with lowered accuracy are also given in Figure 5.13. The individual sets are described by their unique pair of vehicle width and length. One can see from the comparison of Figures 5.10 and 5.13 that the number of anonymity sets significantly decreases with increased discretization steps for vehicle dimensions. However, even a discretization step of 0.5 m leads to some sets with a (very) small share on the overall set of vehicles. Hence, nodes from those sets cannot expect a high level of privacy.

Results on VU for a system using $|e_i| = 2$ together with lowered accuracy of node's dimensions are given in Figure 5.14. The comparison of Figures 5.11 and 5.14 shows that lowering the resolution of vehicle dimensions, as given in CAMs, significantly decreases VU for most vehicles. Hence, privacy of nodes is significantly increased. Discretization steps of 0.2 m (best 0.2 / average 0.2) and 0.5 m (best 0.5 / average 0.5) are considered to achieve the results given in Figure 5.14. The most common group of vehicles holds about 20.8% and 50.8% of all nodes for resolutions of 0.2 m and 0.5 m, respectively (see also Figure 5.13). Unfortunately, the worst case behavior is unchanged in comparison to smaller discretization steps. These findings are in line



(a) CDF of anonymity sets resulting from e_i with a commonly used AAC and vehicle dimensions' resolution reduced to 0.2 m.



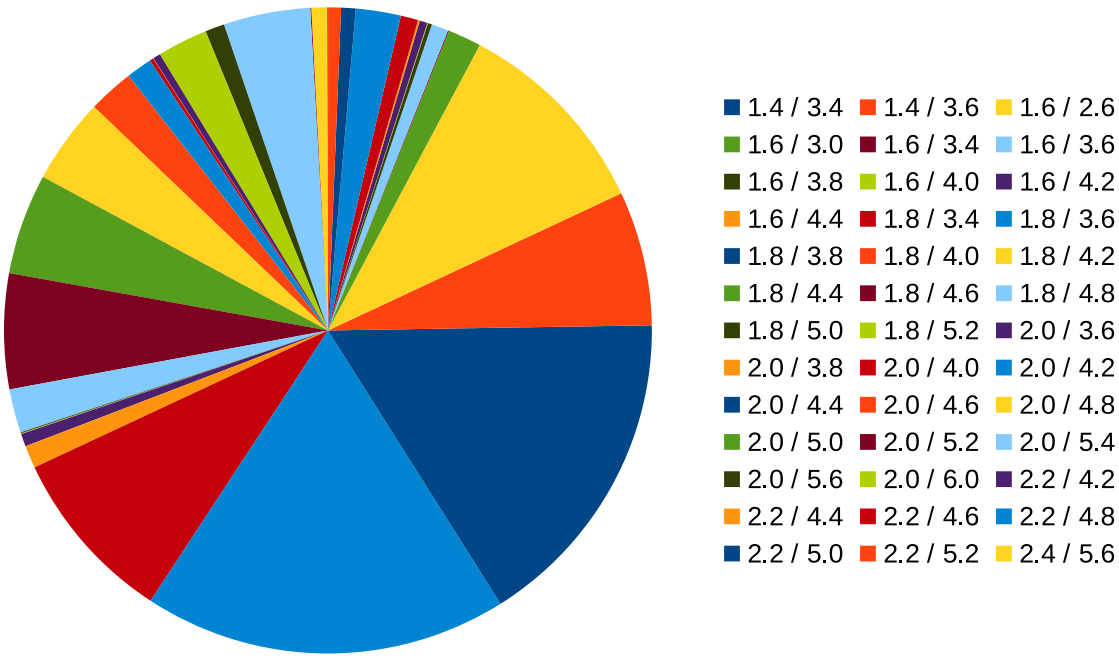
(b) CDF of anonymity sets resulting from e_i with a commonly used AAC and vehicle dimensions' resolution reduced to 0.5 m.

Figure 5.12: CDF of anonymity sets resulting from $|e_i| = 2$ and lowered vehicle dimension's accuracy.

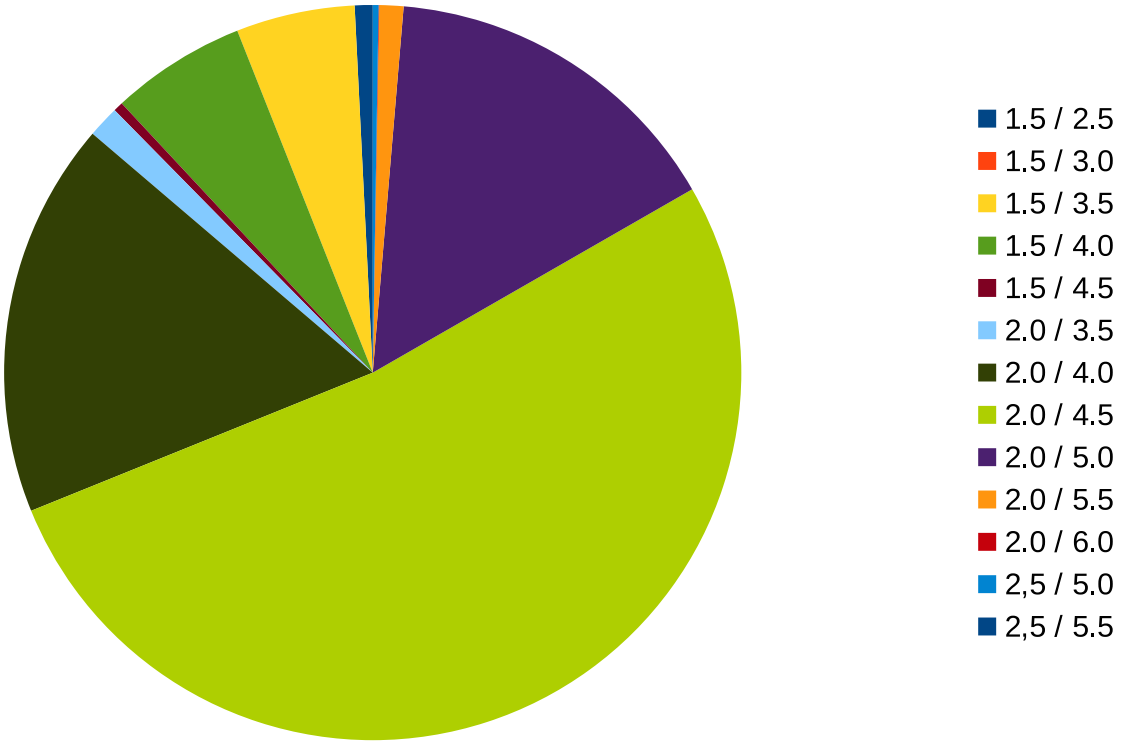
with the findings regarding the distribution of anonymity sets for lowered vehicle dimensions' accuracy discussed before.

Usage of a lowered node dimension's resolution without a common AA would be possible. However, we find that the increase in vehicle privacy is quite low even when using a 0.5 m resolution. The reason for this is that within the fleet of a single OEM, there is a far smaller set of nodes with whom a node can be identical regarding its dimensions, in comparison to the set of all nodes from all OEMs. Thus, we discourage to use the discretization approach only.

Summary of Evaluation The obtained results show that even without other tracking mechanisms, an attacker can perform node tracking with high probability using just a small set of characteristic constant data, even though the node performed a pseudonym change. This shows that the presence of characteristic constant data is able to render pseudonym changes useless, as an attacker can simply re-identify nodes after the pseudonym change. Combining this attack with further tracking mechanisms, e.g., from [303], promises to achieve even higher tracking probabilities. Thus, the suggested precautions for avoiding characteristic constant data in VANET data sets on all protocol layers should be used to limit the traceability of vehicles. Thereby, the level of privacy for drivers is enhanced significantly.



(a) Anonymity sets resulting from 0.2 m discretization steps used for vehicles' dimensions.



(b) Anonymity sets resulting from 0.5 m discretization steps used for vehicles' dimensions.

Figure 5.13: Anonymity sets from node dimensions in CAMs with lowered accuracy.

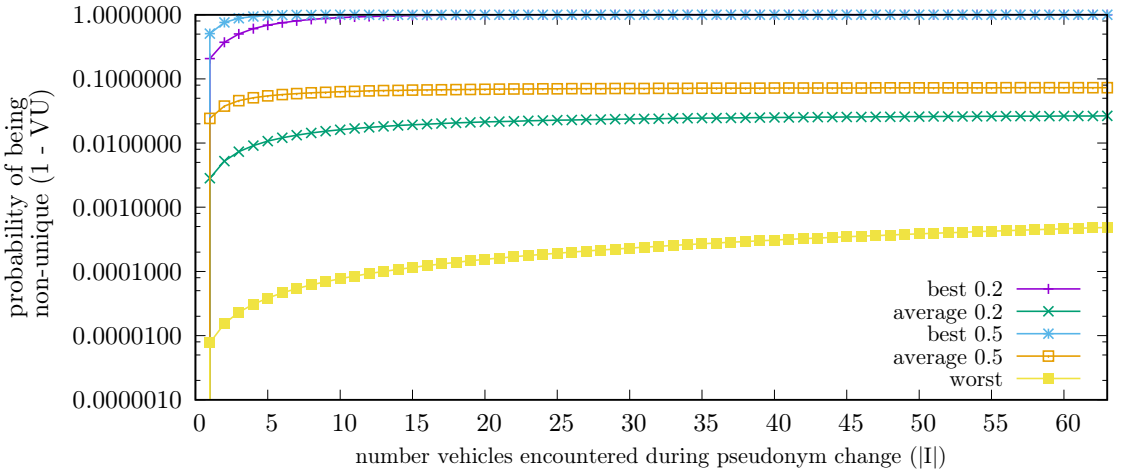


Figure 5.14: Vehicle uniqueness during pseudonym change with $|\boldsymbol{e}_i| = 2$ and lowered vehicle dimension's accuracy.

5.5 Summary of Requirements Emerging from Advanced Attacks

Several security issues have been identified in prior sections of this chapter, which emerge from the obtained advanced attacks on VANETs. These include,

1. secure time synchronization between nodes is required, but state of the art mechanisms can hardly provide it,
2. secure obtaining of a nodes absolute position is required, but there is a lack of mechanisms to provide this information,
3. presence of multiple PSCs within a node being valid during the same time interval should be avoided,
4. pre-caching of PSCs valid in the future within nodes should be limited to a minimum,
5. presence of constant but distinctive data sets within VANET messages, so called characteristic constant data, has to be avoided to enable privacy preserving pseudonym changes,
6. mechanisms for limiting the channel load during certificate chain distribution are required, especially the number of certificate chain deliveries after a request for a CA certificate should be limited to a minimum,
7. the VANET communication channel needs to provide enough spare capacity to ensure the fulfillment of minimum cooperative requirements of applications in case nodes include their PSC in every single message.

Approaches to fulfill the given requirements, especially for ETSI ITS, are given in Chapter 6.

Chapter 6

Certificate Handling in VANETs

A number of requirements for efficient and secure certificate (chain) handling are identified in Chapters 4 and 5. In the following, proposals for improvements to standardized ETSI ITS and WAVE systems are suggested in order to make these systems fulfill the newly found requirements. In doing so, the discussion is separated into

- improved PSC dissemination provided in Sections 6.1 and 6.2,
- a new approach for CA certificate distribution given in Section 6.3,
- improved PSC refill mechanisms discussed in Section 6.4, and
- a novel proposal to limit the impact of PSC change on neighborhood aware PSC dissemination introduced and evaluated in Section 6.5.

6.1 Adaptive Situation Aware Cyclic Pseudonym Certificate Distribution

The standardized PSC distribution mechanism is described in detail in Section 2.2.4.3. One out of three parts of this strategy is to distribute a node's PSC in a cyclic manner. However, only fixed cycle times have been considered in prior work. To further improve the effectiveness and efficiency of the standardized algorithm, adaptation of the PSC inclusion frequency is considered in the following. The proposed algorithm yields to provide at least the same low level of cryptographic packet loss as the standardized one while lowering the channel load by less frequent PSC dissemination. A side effect of a lowered PSC inclusion frequency is to increase the amount of data within messages usable for higher protocol layers, e.g., the facility layer. This is caused by the mutual dependency of variable length data fields included by different protocol layers, as identified in Section 4.3. Topics treated within this section are partly covered by prior work of the author in [33][1].

[1] Contribution of the co-authors is mainly related to implementation of traffic scenarios and the outlined algorithm within the used simulation environment. The main contribution is from the author of this work.

6.1.1 Algorithm Design

The basic idea of the adaptive PSC distribution scheme proposed in the following is to adapt the frequency of cyclic PSC emission to the currently experienced vehicular environment. In general, the new algorithm is built on top of the standardized mechanisms from [125]. These are parametrized as recommended in Section 4.2.1. Hence, implicit and repeated explicit PSC requests are used in combination with cyclic PSC emission. However, the distribution algorithm from [125] is changed in regard to the following major points.

1. Position-based weighting of a request's significance is applied. In prior work all requests are weighted equally.
2. The PSC inclusion frequency is varied based on the accumulated weights of received requests.

To determine the significance of a request, its sender get assigned to one out of four relevance areas. This assignment is based on the sender's relative distance to the receiver. This concept is illustrated in Figure 6.1.

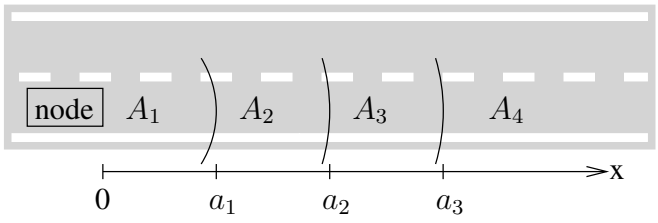


Figure 6.1: Significance areas around a vehicle.

Discretization of a node's environment is inspired by the evaluation concept used in [133, 135]. However, [133, 135] use this concept only for offline evaluation with global knowledge about the whole network, i.e., in connection with an available ground truth. In contrast, these areas are used for online calculation of a metric representing a node's environment in the following. Moreover, sizes of the different relevance areas are varied based on the current communication conditions, while work in [133, 135] uses areas of a-priori fixed size.

The boundaries a_i ($i \in [1; 4]$) of the individual areas A_i, as shown in Figure 6.1, are given by Equations 6.1 to 6.4.

$$a_1 = \frac{1}{N} \sum_{j=1}^{N} d_j \tag{6.1}$$

$$a_3 = \max d_j; j \in [1; N] \tag{6.2}$$

$$a_2 = \frac{a_1 + a_3}{2} \tag{6.3}$$

$$a_4 = \infty \tag{6.4}$$

N gives the number of currently known nodes in the node's surrounding. The distance between the own node and (another) node j is denoted by d_j. A node is removed from the list of known

nodes, if no message from it has been received within a fixed time span (timeout). A timeout limit of two seconds is used in the following. It is selected to correspond to the doubled maximum transmission interval of CAMs. Thus, the algorithm tolerates missing at least one CAM from another node without removing it from the list of known nodes. The fourth area A_4 is used to filter requests from nodes, which are so far away that no reliable communication with them is possible, i.e., only sporadic message exchange happens. [133, 135] uses fixed values of $a_1 = 100m$, $a_2 = 200m$ and $a_3 = 300m$.

After a request from relevance area A_i was received, the current authentication ratio r_i inside A_i is determined by

$$r_i = \frac{n_{i,auth}}{n_{i,known}};\ n_{i,auth} \leq n_{i,known};\ r_i \in [0;1]\,. \tag{6.5}$$

With $n_{i,auth}$ giving the number of nodes within A_i whose PSC is known and verified, i.e., these nodes are authenticated. $n_{i,known}$ gives the number of all known nodes within A_i, i.e., such nodes from whom at lest one message has been received within the forgetting timeout interval described above.

The different authentication ratios r_i get combined to a unified weighted authentication ratio r_w by

$$r_w = \sum_{i=1}^{3} w_i \cdot r_i;\ \sum_{i=1}^{3} w_i = 1;\ w_i \geq 0. \tag{6.6}$$

Thus, r_4, i.e., the authentication ratio within A_4, is ignored for determining the PSC emission frequency. This is done as communication with nodes inside A_4 is regarded as unstable and of minor importance, especially in comparison to communication within areas being closer to the monitoring node (A_1 to A_3).

In general, communication with nodes in the close environment of a node is considered more important for safety critical applications, like collision avoidance, in comparison to nodes being further away. Hence, we recommended to use the criterion $w_1 > w_2 > w_3$ in the selection process of weights w_i. Thereby, the influence of unauthenticated nodes in A_1 higher than the ones in A_2, which have more impact than nodes in A_3. The used selection process for parameters w_i is described after the introduction of the remaining algorithm.

The current time interval t_{cert} between two successive PSC emissions is determined via

$$t_{cert} = \begin{cases} \max\left[\left(\frac{r_w}{1-r_w}\right)^z \cdot t_{cert,min}; t_{cert,min}\right] & r_w < 1 \\ \infty & r_w = 1 \end{cases}. \tag{6.7}$$

The PSC inclusion frequency f_{cert} is given by $f_{cert} = t_{cert}^{-1}$. Furthermore, $z \geq 0$ holds. Equation 6.7 is chosen in a way that t_{cert} varies alongside with r_w, and the scaling factor $\left(\frac{r_w}{1-r_w}\right)^z$ of the minimal PSC inclusion period $t_{cert,min}$ may have arbitrary values in the range between 0 and ∞. This overcomes the fixed, standardized setting of t_{cert} from [125, 176]. Replacing the scaling factor with a fixed number yields a system like specified in current ETSI ITS and WAVE standards. The relation between t_{cert} and r_w is illustrated in Figure 6.2.

In case of $r_w = 1$, cyclic inclusion of certificates is turned off, i.e., $f_{cert} = 0$. This is done, as $r_w = 1$ relates to a system status in which the node does not know about any other

unauthenticated node within its surrounding A_1 to A_3. Hence, further dissemination of the PSC in a cyclic manner is considered to be pure overhead. Another station being in need of the PSC can still obtain it via an explicit PSC request, which triggers sending of the PSC even in case cyclic dissemination is turned of.

The minimum value of t_{cert} ($t_{cert,min}$) is given by the minimum time interval between sending of two CAMs. The lower limit for $t_{cert,min}$ ($\min(t_{cert,min})$) is given by the 10 Hz maximum CAM emission frequency, i.e., a period of $\min(t_{cert,min}) = 0.1\text{s}$. This determines the maximum PSC emission frequency, as the security entity cannot trigger the sending of messages on its own, but relies on piggybacking its data to messages generated at higher protocol layers, like CAMs. The parameter z is used to adjust the reactivity of the algorithm to changes in the monitored weighted authentication ratio in its surrounding.

The influence of the parameter z on the inclusion period of PSCs is shown in Figure 6.2 for the case of a CAM emission frequency of 10 Hz ($t_{cert,min} = 0.1\text{s}$). One can see from

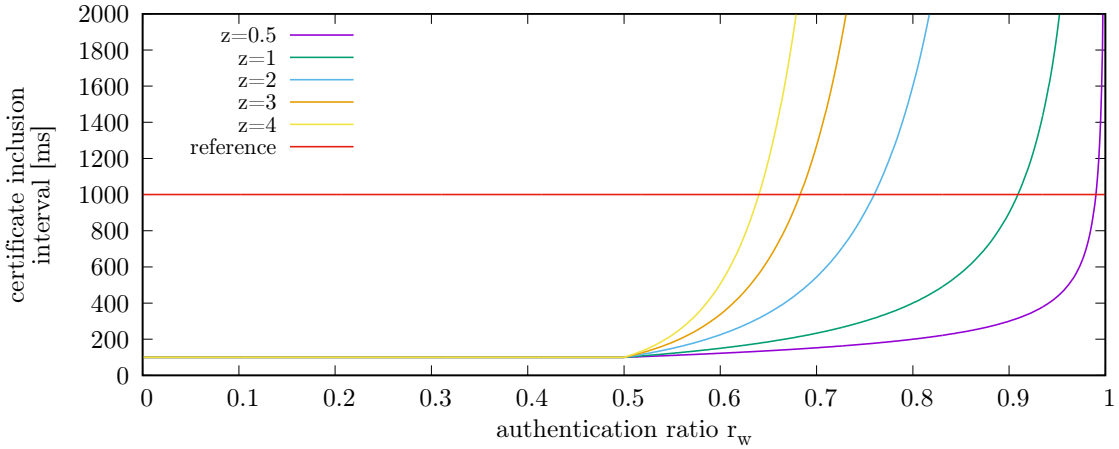


Figure 6.2: Influence of parameter z on the PSC inclusion interval.

Equation 6.7 and Figure 6.2 that for $r_w = 1$ cyclic PSC emission is turned off. This corresponds to a traffic scenario in which the surrounding of a node does not change over time, e.g., inside a large scale traffic jam. In such kind of traffic scenarios there is no need for PSC emission, as all nodes already know about the PSCs of nodes within their communication range. An approach using $z = 0$ is looked at in Section 6.2.

Decreasing values of z lead to increased changes of t_{cert} alongside changes in r_w, as illustrated in Figure 6.2. Hence, reaction of the PSC emission algorithm on detected changes in a node's surrounding is faster for lower values of z. However, this may lead to an overreaction, as it takes time until feedback (from a CAM with included PSC) arrives from the node(s) causing $r_w \neq 1$. During that time interval unnecessary PSC emissions may occur, due to a too strong reduction in t_{cert} for very low values of z, i.e., $z << 1$. This shows the need to consider the trade-off between channel load and cryptographic packet loss, i.e., discarded received packets due to not available PSCs for their verification.

The reference value shown in Figure 6.2 is the fixed cyclic PSC inclusion interval of 1 s from [125]. The adaptive scheme uses a significantly longer inclusion interval for high values

of r_w, which can be expected to lower channel utilization within a well known neighborhood.

Simulation based testing of different parameter combinations for w_i and z within the proposed adaptive PSC distribution strategy was used to select the values of $w_1 = 0.6$, $w_2 = 0.3$, $w_3 = 0.1$ and $z = 0.5$. These were found to provide the best performance in regard to channel utilization and cryptographic packet loss. Thus, these parameters are used in the following.

One should note that currently the security envelope of CAMs does not hold a location stamp. However, this information is present in a required data field of a CAM [119, 125]. Thus, the implementation used for evaluation of this approach looks into the secured data to obtain this information. This is not required for BSMs, as their security envelope includes a location stamp [176].

6.1.2 Evaluation of Adaptive Pseudonym Certificate Distribution

Evaluation of the proposed adaptive PSC dissemination strategy uses the methodology described in Chapter 3. The standardized PSC emission strategy is used as a reference scheme. It is parametrized with repeated explicit PSC requests, as this setup is found to perform best in Section 4.2.1.

At first, the emission rate of PSCs is looked at. This criterion directly influences the average size of sent messages. Thus, a higher PSC emission rate leads to increased channel load for the same amount and timing of sent messages.

Figure 6.3 displays results for the freeway scenario (see also Section 3.2). These show that for most traffic densities the average rate of PSC emission is lower for the proposed adaptive scheme in comparison to the standardized one. However, the difference is quite small and there is a large overlap of measured standard deviation intervals.

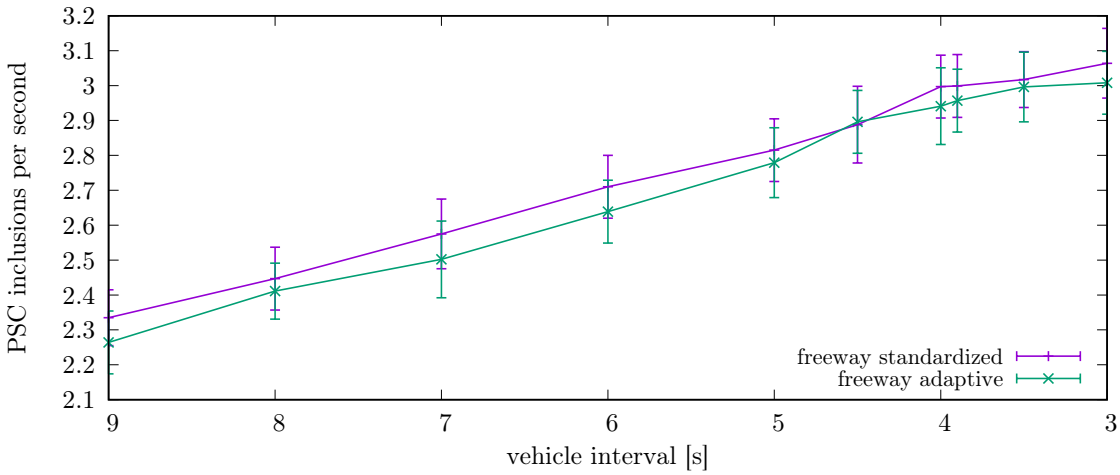


Figure 6.3: PSC emission rate for a freeway scenario under adaptive PSC emission.

Obtained results for the urban roundabout scenario (see also Section 3.2) are given in Figure 6.4. The given results show that the adaptive scheme significantly outperforms its standardized counterpart. For all considered traffic densities, the average PSC emission rate is re-

markably lowered by using the proposed adaptive scheme. The difference is larger for lower traffic densities, which can be expected, as in such traffic scenarios new neighbor detection happens less frequently than in scenarios with higher node density. Thus, cyclic PSC distribution being influenced by the proposed adaptive scheme has a more significant role in the overall PSC distribution procedure for lower density scenarios. In contrast, in high density scenarios the overall PSC dissemination strategy is governed by PSC emission caused by new neighbor detection, which is a common sub-strategy of the standardized and adaptive schemes.

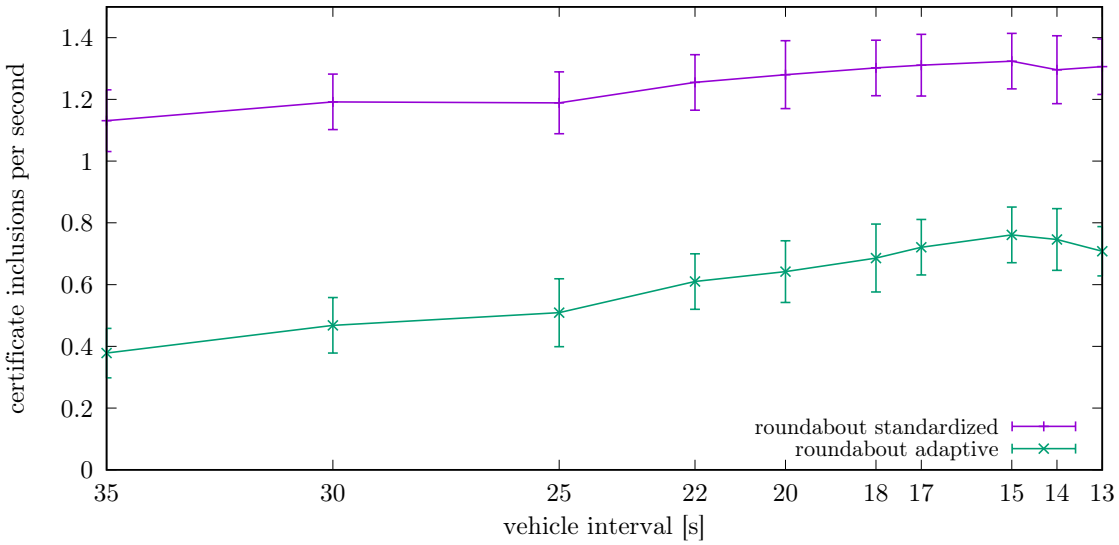


Figure 6.4: PSC emission rate for a roundabout scenario under adaptive PSC emission.

Figure 6.5 gives achieved results for the rural road scenario. One can see that the proposed adaptive algorithm significantly reduces the number of PSC transmissions per second. Hence, less bandwidth is consumed by PSC dissemination in comparison to the standardized PSC distribution scheme. The gain is smaller than in the urban roundabout scenario (lower node mobility), but higher than in the freeway scenario (higher node mobility).

Results for the urban grid scenario are provided in Figure 6.6. One can see that the adaptive scheme slightly outperforms the standardized one.

Comparison of Figures 6.4 and 6.6 shows that average PSC emission rates for these two urban scenarios vary significantly. This is caused by the different traffic shapes in these scenarios. Within the urban grid scenario the frequency of nodes entering and leaving each others communication range is significantly higher than in the urban roundabout scenario. Hence, new neighbor detection happens more frequently and PSC exchange has to be performed more rapidly.

The number of unknown PSC-IDs received in the freeway scenario is almost the same for the adaptive and standardized approaches, as shown in Figure 6.7. These kind of messages receptions lead to cryptographic packet loss. Hence, the adaptive scheme requires a lower amount of PSC emissions (see Figure 6.3) to achieve the same low amount of discarded messages. The given results are collected for nodes within A_1 and A_2.

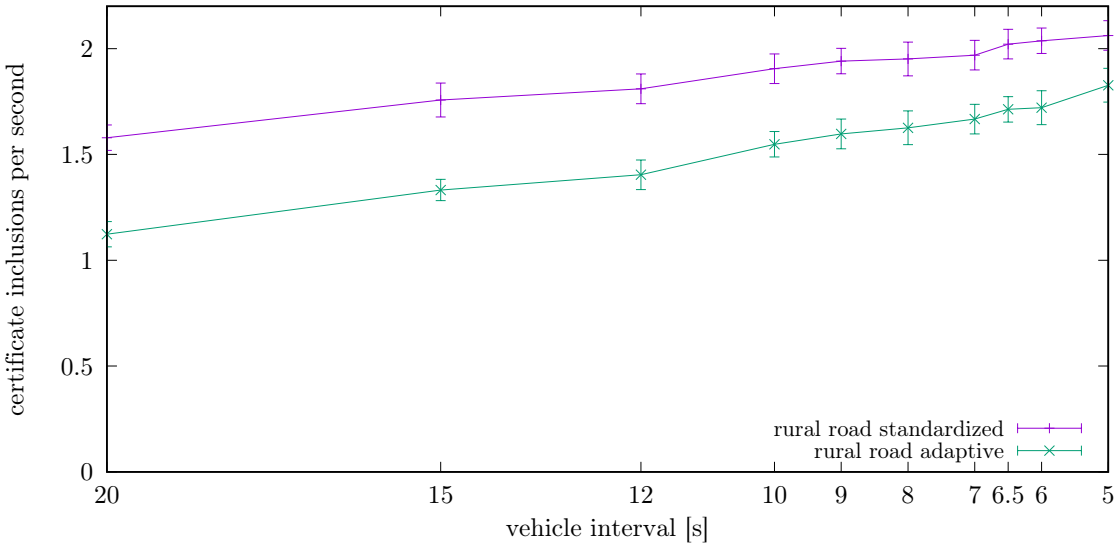


Figure 6.5: PSC emission rate for a rural road scenario under adaptive PSC emission.

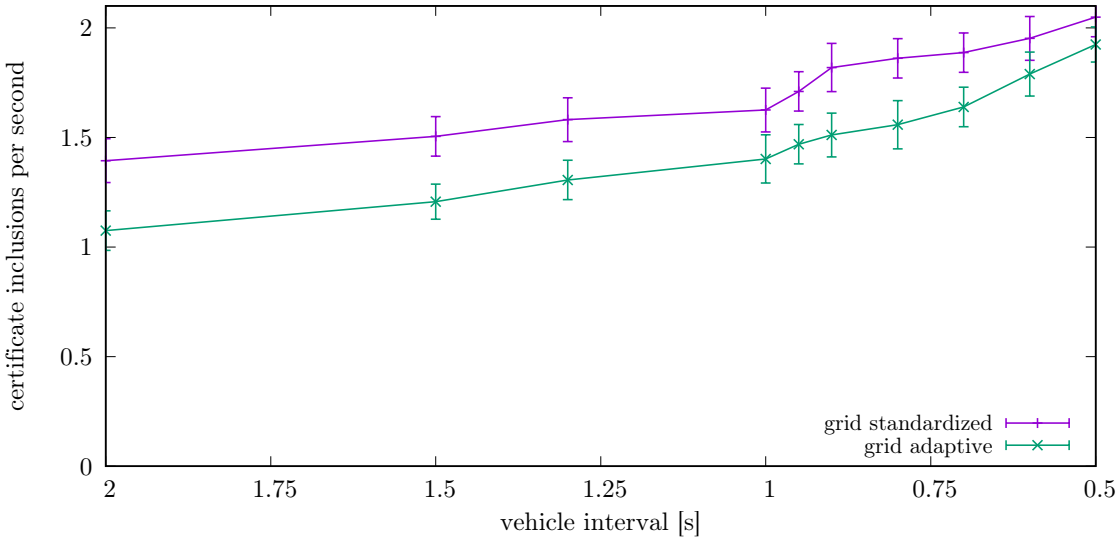


Figure 6.6: PSC emission rate for an urban grid scenario under adaptive PSC emission.

Figure 6.8 displays the results on unknown PSC-IDs in the rural road scenario. The displayed results show that there is no statistically significant performance deviation of both considered PSC emission schemes. There is an almost full overlap of obtained standard deviation intervals, and averages are very close to each other. Hence, the adaptive scheme uses a lower PSC emission rate (see Figure 6.5), but achieves the same low amount of cryptographic packet loss than the standardized mechanism.

Corresponding results for the roundabout scenario are given in Figure 6.9. In this scenario the adaptive scheme is able to further reduce the already very low rate of discarded messages.

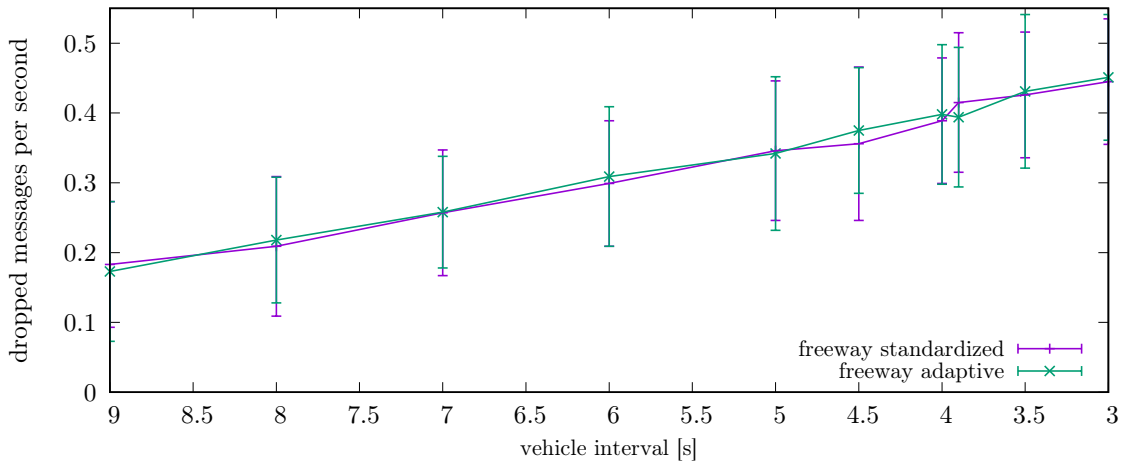


Figure 6.7: Cryptographic packet loss in the freeway scenario.

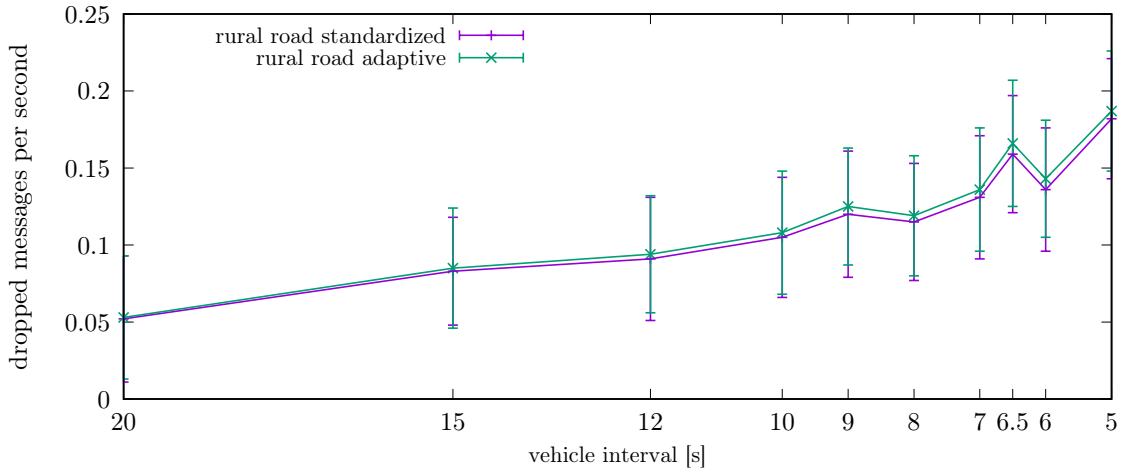


Figure 6.8: Cryptographic packet loss in the rural road scenario.

For all traffic densities the number of lost messages is hardly noticeable. This means that PSC distribution can almost completely avoid cryptographic packet loss.

Results on cryptographic packet loss for the urban grid scenario are displayed in Figure 6.10. The difference between obtained values for the adaptive and standardized approach is quite small. Regarding averages, the standardized scheme slightly outperforms the adaptive one for lower traffic densities, while the converse holds for high traffic density. Like for PSC emission, a significant difference between the results for both urban scenarios is found.

In summary, one can state that the proposed situation aware extension of the standardized PSC distribution scheme yields better system performance. It lowers PSC emissions rates causing decreased channel load, while yielding about equal characteristics of cryptographic packet loss, or even lower this negative impact of sporadic PSC emission in some traffic scenarios.

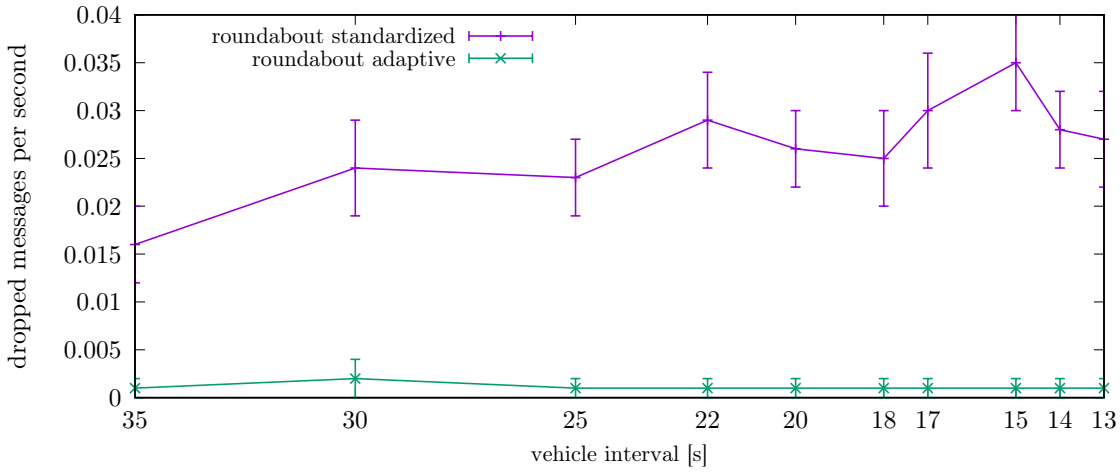


Figure 6.9: Cryptographic packet loss in the roundabout scenario.

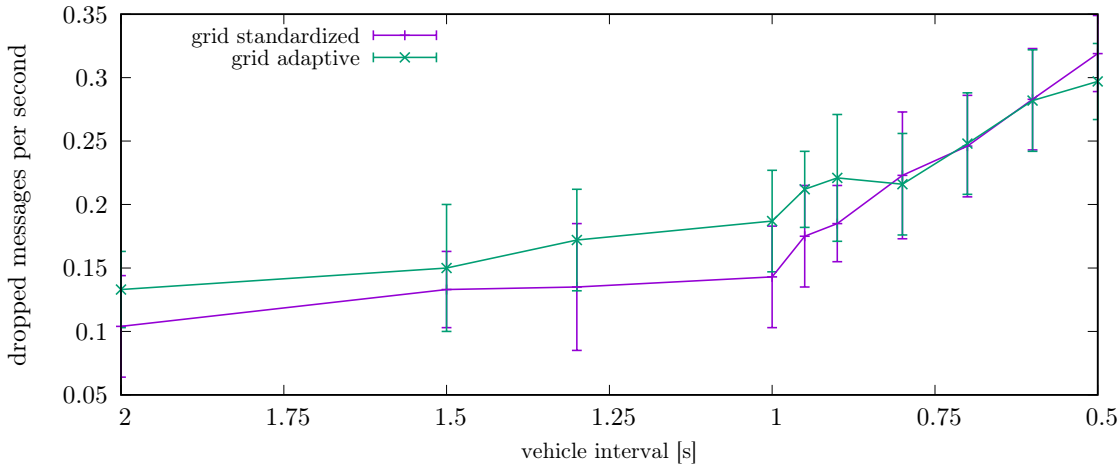


Figure 6.10: Cryptographic packet loss in the urban grid scenario.

6.2 Certificate Distribution via Bursts for Low Mobility Scenarios

The suggested burst based PSC dissemination scheme from this section is a special case of the adaptive scheme from Section 4.2, as also treated in [31][2]. In this case, $z = 0$ holds, which changes Equation 6.7 to 6.8.

$$t_{cert} = \begin{cases} t_{cert,min} & r_w < 1 \\ \infty & r_w = 1 \end{cases}. \tag{6.8}$$

Moreover, r_w only includes data from A_1 and A_2, i.e., unknown nodes from A_3 and A_4 are ignored (see also Equations 6.1 to 6.4). This means that in case an unauthenticated node is

[2]Contribution of the co-authors is mainly related to implementation of traffic scenarios and the outlined algorithm within the used simulation environment. The main contribution is from the author of this work.

known to be in the close vicinity (A_1 and A_2) of a node, its PSC is always included. The maximum temporal length of the burst created by a single unknown node is limited by a timeout interval, after which the unauthenticated node is removed from the list of nodes whose presence is known. Moreover, the burst is immediately ended in case all nodes within A_1 and A_2 become authenticated nodes. A timeout of one second is used in the following. This ensures that the PSC is sent at least twice in case of 1 Hz CAM emission and failure to deliver the PSC during its first emission following new node detection within A_1 or A_2. The limitation to A_1 and A_2 is motivated from the targeted low mobility scenarios, which offer more time for reaction of ADAS to behavior of nodes being pretty far away from the ego node.

The performance of the proposed PSC emission scheme for the urban roundabout scenario is shown in Figure 6.11. The displayed results show that in average the burst based approach slightly outperforms its adaptive counterpart. However, considering the standard deviation intervals, the difference in performance has to be considered statistically insignificant.

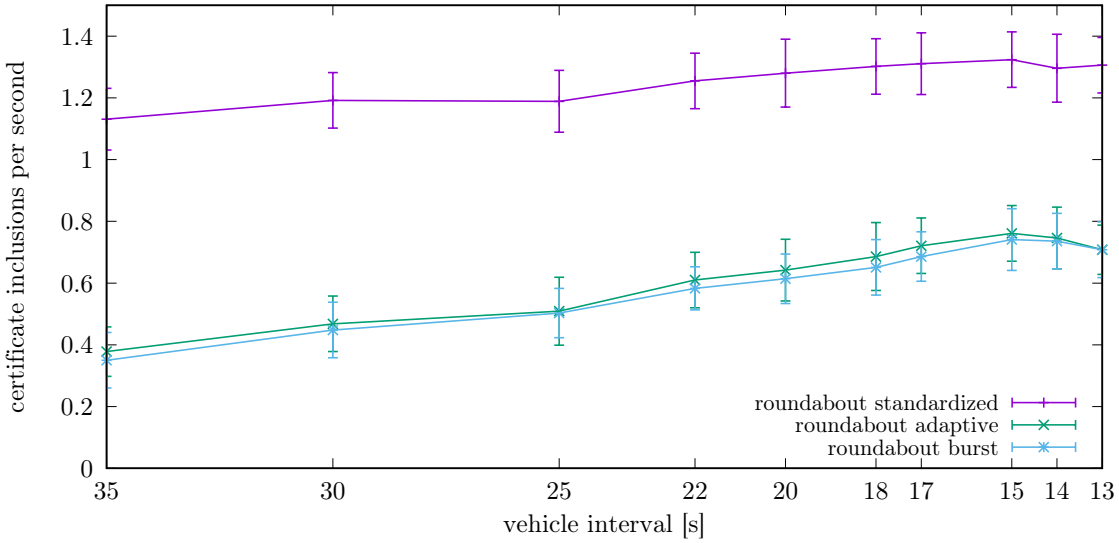


Figure 6.11: PSC emission for an urban roundabout scenario under burst based PSC emission.

The negative impact of burst based PSC emission in a high mobility scenario is illustrated in Figure 6.12 using a freeway scenario. One can clearly see that the PSC emission frequency is much higher for this scheme in comparison to the ones from the standard and from Section 4.2.

The given results also show that selecting very small values for z, i.e., close to zero, causes the approach from Section 4.2 to overreact to newly detected nodes, which are temporarily unauthenticated. Hence, the selection of z is a trade-off between fast (enough) reaction, i.e., z should be small, and avoiding overreactions, i.e., z should not be too small.

In summary, one can state that the burst based PSC emission scheme works well for low mobility scenarios like the urban roundabout scenario. However, the gain is small in comparison to the adaptive approach from Section 4.2, which is shown to work well for all considered scenarios. Hence, we recommended to use the approach from Section 4.2 to avoid extra complexity by a mobility aware selection of the PSC distribution algorithm.

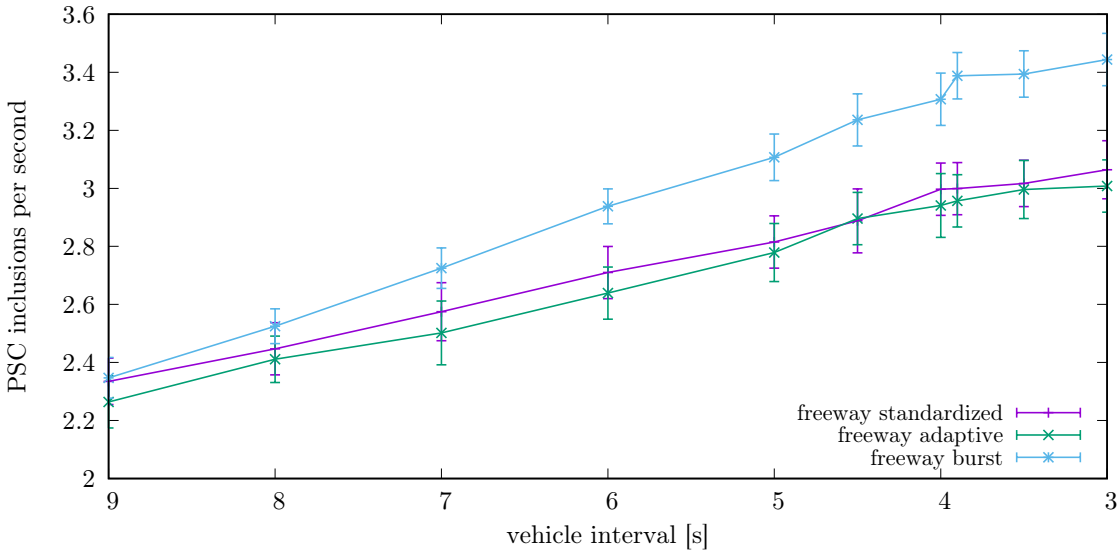


Figure 6.12: PSC emission for a freeway scenario under burst based PSC emission.

6.3 Certificate Chain Distribution

The need for efficient distribution of entire certificate chains has been shown in Sections 4.2.2 and 5.1.2, alongside with drawbacks of the currently standardized simple explicit request based mechanism. Hence, a proposal for an advanced distribution strategy for the dedicated elements of a certificate chain is proposed in the following, which is partly given in the author's prior work in [4, 39, 41][3], too. The discussion is separated into two parts, which are

1. approaches to avoid multiple delivery of CA certificates to

   (a) limit the bandwidth consuming response to an AAC request, which also

   (b) avoids the DOS vulnerability found in Section 5.1.2.

   Such kind of approaches are discussed in Section 6.3.1.

2. A proposal for extending one of the approaches from Section 6.3.1 towards a system only distributing single dedicated certificates is given in Section 6.3.2. This allows to significantly limit the worst case size of the security envelope, as asked for in Section 4.3.

6.3.1 Avoiding Multiple Delivery of a Certificate Authority Certificate

Instead of multiple delivery of an AAC towards its requester, as done in the currently standardized approach (see Section 4.2.2), it is sufficient to deliver the AAC only once. Such kind of scenario is illustrated in Figure 6.13. In the given example, only one out of three possible responders answers the request, although nodes B,C and D are assumed to use the same AAC. In

[3] See also footnote 4.

contrast, all nodes, i.e., B, C and D, should send the AAC according to the currently standardized behavior as illustrated in Figure 2.9.

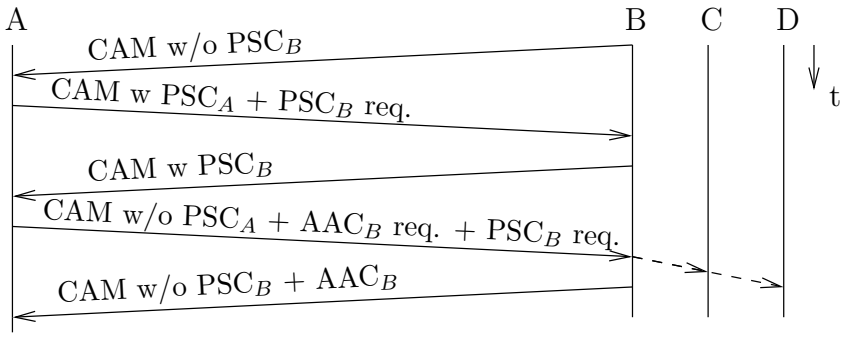


Figure 6.13: AAC request and delivery mechanism with only a single responder.

One should note that the concerned AAC delivery problem shows similarities to the packet forwarding problem encountered in the domain of position based routing. One part of this problem is that the number of forwarders should be minimized to limit the channel load. The main difference to AAC distribution is that the source of the request (for forwarding) is different from the target of the delivery, while in the AAC distribution case the requester is identical to the target of data delivery.

To minimize the number of responses to a request for an AAC, multiple approaches can be thought of. These can be grouped into

1. explicit selection of a dedicated responder at the requester, and
2. decentralized responder selection among all nodes being able to answer the request.

Approaches from both groups are well known in the position based routing domain. Members of group no. 1 typically try to maximize/minimize one or multiple parameters of the communication setup. Thus, they are commonly called greedy (forwarding) schemes. Hence, we call such approaches for AAC delivery *greedy responding* schemes. They are discussed in Section 6.3.1.1. A popular scheme from group no. 2 is contention-based forwarding (CBF). Therefore, we use the term contention-based responding (CBR) for algorithms using such kind of decentralized responder selection for AAC delivery. An analysis of CBR is provided in Section 6.3.1.2.

6.3.1.1 Greedy Responding

There are mainly two criteria for optimization of the requester based responder selection process. These are to either

- minimize the distance of the chosen responder to the requester, which corresponds to an approach maximizing the probability of successful message exchange, or to
- minimize the time until the responder answers the request. In doing so, one has to keep in mind that responses do not trigger message sending on their own, in contrast to message

forwarding, but rely on piggybacking on beacon messages generated by higher layers, e.g., CAMs.

One could also think of combinations of time and position based criteria in the responder selection process. However, such approaches are not considered in the following, as the conducted evaluation already shows well usability of the simple, and thus easy to implement, approaches.

Position Based Selection The requester can try to maximize the probability for successful bidirectional communication, including request and response. In doing so, different strategies can be thought of. These include a

- simple strategy using only the positions of available responders, and
- advanced strategies using an environment model of the requester.

Required data like position, speed and heading of nodes is contained in corresponding beacon messages, i.e., CAMs or BSMs. For the simple strategy, a requester minimizes the distance between requester and responder. Thereby, it tries to maximize chances that the selected node actually receives the request, and its reply is successfully delivered to the requester, too. This strategy is based on the assumption that the probability of two nodes successfully exchanging data increases with decreasing distance between these nodes. An evaluation of this approach is given in Section 6.3.3. In WAVE, the required position stamp is contained in the security envelope of beacons, while in ETSI ITS the security envelope does not contain such data. Hence. the implementation used for evaluation takes this information from the facility layer CAM data, which is stored in the security envelope's payload. The logically corresponding approach from packet forwarding is to maximize the progress towards the final receiver by each forwarding hop.

The simple approach does not guarantee to answer the request in minimal possible time. Time to delivery of the AAC ($t_{delivery}$) is determined by the (current) beacon generation intervals of both the requester ($\Delta t_{beacon,requester}$) and the responder ($\Delta t_{beacon,responder}$), due to piggybacking of both request and response on CAMs. Therefore,

$$t_{delivery} \leq \Delta t_{beacon,requester} + \Delta t_{beacon,responder} \tag{6.9}$$

holds. Hence, in ETSI ITS it can take up to two seconds until the AAC request gets answered (WAVE: 200 ms). High mobility of nodes may cause the situation that the chosen responder is no longer the closest possible responder, when answering the request. However, the simple strategy provides the benefit of being easy to implement.

Advanced strategies could use a model of the communication conditions within a requester's surrounding. Approaches for such models are typically based on accurate digital maps and monitoring of other nodes, like proposed in [48]. However, real time maintenance of such models is still not possible, due to very high computational requirements. Thus, this approach is not considered in the following.

A long time span between requesting and arriving of the response poses the drawback of possible extra cryptographic packet loss, as the AAC in need is shared by many nodes. To reduce the chance of a long time span until AAC delivery, the following strategy uses the next expected beacon transmission time as the main criteria to select the responder.

Sending Time Based Selection The requester of an AAC can minimize the time span Δt, until the requested AAC is delivered in case no package loss occurs. In VANETs with fixed beaconing intervals, like WAVE, the requester can directly calculate the next beacon transmission time of all known nodes. This calculation is based on the sending time stamp from within the security envelope [125, 176]. However, in ETSI ITS the CAM generation interval varies [119].

A node's current CAM generation interval is mainly determined from vehicle dynamics, e.g., speed or turn rate. Fortunately, these data sets are contained in CAMs together with the current generation interval. One can assume that vehicle dynamics are quite constant in the short time span between emission of two CAMs. Hence, a receiver can determine a quite accurate hypothesis about the next CAM's generation time. However, DCC inducted limitation of CAM generation may lead to a less accurate hypothesis, in case such limitations change rapidly.

6.3.1.2 Contention Based Responding

For the case of CBR the two cases of position and time based responding, which is close to CBF, and the case of only timeout based responding are considered in the following.

Position and Timeout Based Responding The proposals of CBF suggest position and time based selection of message forwarders [140, 141]. The initial sender and (final) target of an AAC request/delivery scheme are identical. In contrast, sender and target are different for multi-hop message dissemination schemes. Thus, the forwarder selection criteria of CBF has to be changed to obtain a suitable responder selection criteria for the CBR concept.

Inspired by the CBF timeout function, the CBR timeout function is chosen to be

$$t = \begin{cases} t_{CAM,i} \cdot \left(1 - \frac{d_i}{d_{max}}\right) & 0 \leq d_i < d_{max} \\ \infty & \text{otherwise} \end{cases}. \tag{6.10}$$

The distance between requester and responder candidate i when receiving the request is denoted by d_i. A maximum distance d_{max} is used to avoid responses to requesters being so far away that only sporadic and unstable communication with them can be expected (see e.g., Section 6.1.1 for a possibility to determine such a boundary). $t_{CAM,i}$ gives the current CAM generation interval at node i when receiving the request. Moreover, as in CBF, a node monitoring the response of another node, cancels its own timeout. Thus, it does not transmit the AAC itself, i.e., no certificate chain emission takes place.

An example scenario for CBR based distribution of an AAC is given in Figure 6.14. The request is received by two nodes, which both are possible responders, as they are assumed to be aware of the requested AAC. Two timeouts influence the distribution process. The first one (right, red in Figure 6.14) is the time until the next message is to be sent. The second timeout (left, green in Figure 6.14) is the one determined from Equation 6.10.

In the chosen example, the most right vehicle is the first one to send a message after the AAC request was sent. However, it is not going to include the AAC into this message, as the timeout from Equation 6.10 will happen after sending the message. Hence, the message will be subject to cryptographic packet loss at the requester, i.e., it gets dropped. Delivery of the AAC is performed by the node next to the requester.

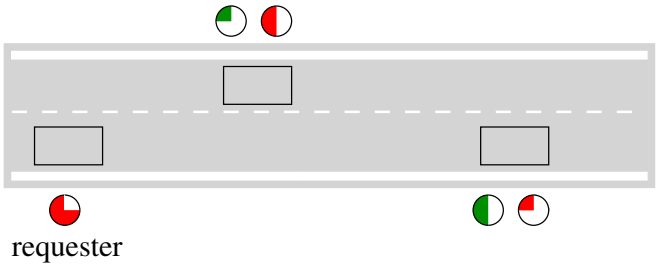


Figure 6.14: Example scenario for CBR based AAC distribution.

This approach tries to minimize the distance between requester and responder d_i. Hence, the set of nodes receiving the response can be assumed to be similar to the set of nodes, which received the request. Therefore, the number of unnecessary extra responses caused by the hidden station problem can be expected to be low.

However, as many vehicles share the same AAC, it is pretty likely that the requester receives more than just one message (e.g., CAMs) whose PSC is signed by the requested AAC. All these messages are discarded (i.e, they lead to cryptographic packet loss), as they cannot be validated. Hence, a strategy trying to minimize such packet loss at the cost of increased probability for duplicate responses is proposed at next.

Pure Timeout Based Responding A simpler variant for decentralized responder selection can use only a responding timeout, i.e., no location information is required in contrast to the above outlined CBR approach. The timeout period is just given by the time until the next beacon message is to be transmitted. Like in the concept proposed above, a node cancels AAC emission when receiving a response from another node.

This concept minimizes the time span until the request is answered. Thus, it also minimizes the probability of additional cryptographic packet loss by discarding messages from other nodes sharing the requested AAC.

A drawback of this simple approach is that the set of vehicles receiving the first response can differ significantly from the set of vehicles, which received the request. Hence, the chance of duplicated replies is higher for this approach in comparison to the one proposed before.

Additionally, the responder could leave the communication range of the requester, before transmitting the response. In the worst case, all other possible responders receive the response. Thus, these nodes suppress their own responses. Hence, the requester does not receive any response. To avoid this scenario, a responder can keep track of its current average communication range. Before sending the response, it checks whether the position of the requester is within this range. Otherwise, it does not send the response. This improvement can also be used for the time and requester based responder selection strategy proposed before.

The pure timeout based CBR concept is illustrated in Figure 6.14 (right timeout). In contrast to position and time based CBR, the most right vehicle will answer the request and the vehicle in the middle will not emit its AAC. Thereby, the CAM following the request can be verified, and does not get dropped, in contrast to the case of the position and time using CBR approach described above.

6.3.2 Removing the Requirement of Certificate Chain Distribution

Explicit selection of a dedicated responder for AAC requests is discussed in Section 6.3.1.1. The following approach to avoid the distribution of certificate chains in a single message is based on this kind of algorithms.

In case a node is selected as the responder to an AAC request, the requester needs to know about the PSC of this dedicated node. Otherwise, the section algorithm is not able to select this node, because knowledge of the PSC is required to know about the AAC used to secure this PSC. Thus, delivery of the full certificate chain of the selected responder is not required, as the PSC contained in this chain is already known to the requester. Thus, its repeated delivery is superfluous and only wastes bandwidth on the wireless channel. Hence, including only the AAC into the security envelope of the response message is sufficient. Thereby, an algorithm is designed for AAC dissemination, which does not need to include entire certificate chains into a single message. This approach shows two major advantages, which are

1. further bandwidth saving for the AAC distribution algorithm, and

2. limiting of the maximum number of certificates present in the security envelope to one. Hence, the worst case size of the security envelope gets reduced in comparison to a system using distribution of a full certificate chain within a single message.

Finding no. 2 is especially important, due to the cross-layer data size dependence found in Section 4.3. A shorter worst case size of the security envelope enables to use a larger application layer payload, while avoiding to violate maximum message size restrictions on the MAC layer, which are enforced by DCC rules.

6.3.3 Evaluation of Improved Certificate Chain Dissemination Schemes

The performance of the suggested certificate chain dissemination schemes from Section 6.3.1 is shown in Figure 6.15. To obtain the given results, the freeway scenario from Section 3.2 is used. The AAC requests are inserted into the simulation as described in Section 4.2.2. The strategy from Section 6.3.2 was found to perform equally to the transmission time based greedy scheme from Section 6.3.1 in regard to the response time. Thus, its results are skipped in Figure 6.15 to avoid overloading it.

Results from Figure 6.15 show that time based responder selection mechanisms outperform their position based counterparts. Performance of the timeout only CBR strategy reaches the one of the standardized approach in almost all considered cases. Thus, the corresponding curves in Figure 6.15 are hardly distinguishable. However, the performance of the proposed time based greedy algorithm is only marginally worse in average than standardized and timeout only CBR approaches. Moreover, the difference can be regarded as being statistically insignificant.

The purely position based greedy responding schemes performs worst. Its performance is partly limited by sporadically selecting nodes with a current CAM generation interval greater than 100 ms. This effect can be expected to disappear in WAVE. Thus, the expected performance would slightly benefit from the fixed 10 Hz beacon interval of WAVE. The CBR scheme using time and position information outperforms the position based greedy scheme, but performs

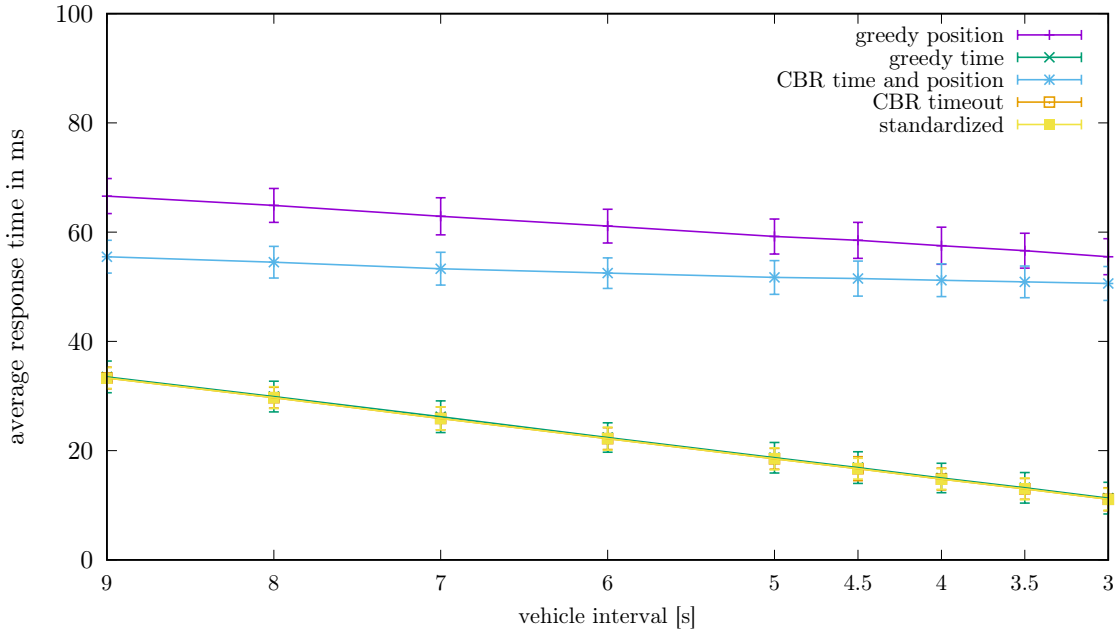


Figure 6.15: Response times of AAC dissemination schemes.

worse than the purely time based schemes. In detail analysis shows that some responses are delayed by the position dependent timeout, which leads to a significant delay of actually carried out AAC inclusion. Moreover, all CAMs of nodes, which skipped transmission of the AAC, are discarded by the requester, as these still cannot be verified, due to the missing AAC. Therefore, position based responder selection schemes are not recommended.

In contrast to the position-based routing problem, no serious drawback of requester based selection schemes in comparison to decentralized responder based selection is found. This is caused by the differing aim of node cooperation between routing and AAC distribution. For AAC dissemination, the target of the caused reaction by the addressed responder is the sender itself. In contrast, for packet forwarding the target is a distant node, which is located out of direct communication range of the sender. Moreover, forwarding triggers message sending on its own, while AAC distribution relies on piggybacking on higher level messages, i.e., CAMs or BSMs.

The time based response algorithms yield minimal cryptographic packet loss. In the ideal case, no other node using the unknown AAC transmits, before the AAC is delivered to the requester. Hence, the requester is not forced to discard further messages after the one initiating the AAC request.

In regard to channel load no statistically significant increase was obtained in the setup using sporadic AAC requests, even in case all nodes use the same AAC. Hence, the impact on VANET communication conditions from on-demand distribution of AACs, with the proposed advanced mechanisms, can be regarded as insignificant. Thus, these mechanisms are recommended for usage in practical VANET realizations.

Moreover, an evaluation of the proposed efficient AAC dissemination schemes as countermeasures to the DOS attack proposed in Section 5.1.2 is conducted. The freeway scenario is

used to evaluate this aspect, too. Obtained results show that all proposed strategies can efficiently avoid both DOS attacks (direct and indirect). The increase in average message size and channel load caused by the attacker is hardly noticeable even in case of frequently repeated AAC requests (each present AAC requested every 100 ms). The maximum observed amount of responders for the decentralized schemes was just two. For the remaining AAC delivery schemes, only one node sent its AAC. Hence, the massive amount of certificate chain emissions provoked for the standardized approach can be clearly avoided by all proposed schemes.

The provided evaluation shows that both proposed time based AAC distribution schemes perform well. However, only the requester based selection scheme (time based greedy algorithm) allows to remove the requirement of transmitting entire certificate chains in one message completely. Thus, this scheme is recommended for future use in VANETs to efficiently distribute certificates of CAs. Generalization of this CA certificate distribution strategy to a general multi-level PKI system is discussed in the following section.

6.3.4 Application to a General Multi-Level PKI System

Prior sections described and evaluated the distribution of CA certificates especially for ETSI ITS. In this VANET approach only one level of CA certificates, namely the AACs, are disseminated on-demand within the VANET. However, the proposed CA certificate dissemination scheme is not limited to such flat CA hierarchies within a PKI.

In case CA certificates from multiple levels of the PKI are distributed in the VANET, the approach from prior sections can be used in an iterative manner. In doing so, the dedicated CA requests are used for each single CA's certificate individually. Each element of the certificate chain, holding multiple CA certificates, can be delivered by a different node, as responder selection is done for each request individually.

The next section considers efficiency of the PSC refill mechanism in ETSI ITS.

6.4 Certificate Refill for Mobile Nodes

Two major challenges exits in the PSC refill process within mobile nodes, i.e., such without a fixed communication to a backbone service. These are

1. enabling PSC refills without a valid PSC being available within a node (see Section 6.4.1), and
2. efficient handling of refill requests at CAs to avoid DOS vulnerabilities of these kind of important backbone services (see Section 6.4.2).

Both issues are studied in the following.

6.4.1 Enabling Multi-Hop PSC Refill Requests

Communication between a mobile node and a backbone CA requires secured multi-hop message exchange. However, there are two major issues in current ETSI ITS standards, which disable such communication. The first problem is the inability of encrypted multi-hop communication,

as outlined in Chapter 6.6. A second problem is the need to sign the PSC request messages at the requester, as outlined in the following.

6.4.1.1 Problem Statement

To enable privacy conserving refill requests, the long term credentials of the requester must not be available to nodes forwarding the request to the CA. Hence, such long term credentials cannot be used to secure the request message at the network layer. Therefore, the requester needs to have access to at least one valid PSC to secure the message. Otherwise, no forwarding of the message will be performed by other nodes, as it does not carry a valid signature [122].

Unfortunately, this requirement of an available valid PSC for backbone connections poses a problem for the PSC usage and refill process. It contradicts with the requirement of not holding PSCs with long validity times (see Section 5.5). This leads to an initialization problem after a node's start-up. A valid PSC is already required for sending a PSC refill request to the corresponding CA. However, a node being inactive for a longer time span, i.e., longer then the lifetime of its last obtained PSC, does not hold a valid PSC, as pre-caching of PSCs is to be avoided. Hence, the node cannot obtain a new PSC from the CA. An approach to overcome this issue is proposed in the following.

6.4.1.2 Initial Pseudonym Requesting via Dedicated Certificates

To solve the initialization problem outlined in the prior section, introduction of a dedicated kind of PSC called PSC_u is proposed. This PSC_u is limited in use for pseudonym refill requests by the help of the ITS-AID field contained in every certificate [125]. Hence, a PSC_u cannot be used to secure any other kind of messages like, e.g., CAMs or DENMs.

The PSC_u is only used to obtain a regular PSC from the CA. After its usage, it is replaced by a new one to protect the requester's privacy. The lifetime of a PSC_u is significantly longer then the one of an ordinary PSC. To provide robustness of the overall VANET implementation, a lifetime in the area of some days is proposed.

An attacker manipulating the internal time of a node (see also Section 5.3.1), can misuse the PSC_u to send bogus requests to the CA. However, the request will hold an incorrect time stamp. Assuming that the attacker cannot manipulate the time base of the CA as well, the bogus requests are dropped by the CA. Hence, the attacker can only create some bogus traffic in the VANET and towards the CA, but cannot inject safety critical messages, e.g., CAMs, into the system like in case of PSCs with long validity times (see also Section 5.3.2).

6.4.2 Efficient PSC Refill Request Handling at CAs

The standardized format for PSC refill request messages is described in Section 2.2.4.4. Handling of the request at the CA includes to first decrypt the request message, and to afterwards check the signature of the message. The meta data within the message holds a unique and unchanging identifier of the long term credentials used to sign the message. It is used by the CA to check the validity of the signer's certificate and to check the digital signature of the request.

Therefore, a CA always needs to decrypt a received request message before verifying its signature. Hence, two computationally expensive cryptographic operations are required before the CA can decide upon the validity of the request. This can be misused by an attacker, who sends bogus messages to the CA to overwhelm its computational resources with the target of performing a DOS attack. The impact of a successful attack would be great, as all nodes assigned to this CA would experience problems in regard to their cyclically required PSC refill routine. Hence, we propose to minimize the attack surface of CA's by the mechanisms introduced in the following. The described approach is also covered by the author's prior work in [30].

To overcome the possible DOS vulnerability a combination of two mechanism is suggested. These include to

1. replace the unique long term credential identifier by a Temporary Identifier (TID), which can be exposed to the public, and to

2. change the sequence of cryptographic operations to first encrypting and then signing the request.

The proposed request message assembling process is illustrated in Figure 6.16. For a comparison to the standardized mechanism see also Figure 2.10. The sequence of cryptographic operations (signing and encrypting) is changed and an additional step for adding the TID is introduced.

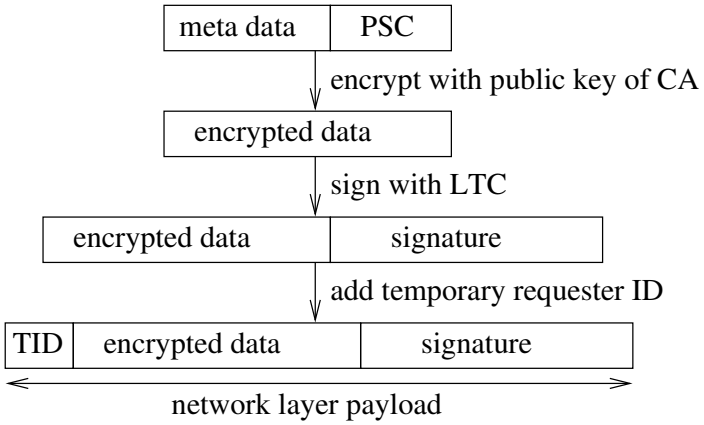


Figure 6.16: Improved format for a pseudonym refill request message.

To protect the privacy of the requester, the TID has to fulfill two criteria. An attacker should not be able to link different monitored TIDs to their common originator, and an attacker must not be able to predict a future TID from such monitored in the past. These properties are fulfilled by some hash chain and rolling code designs, as explained in detail in Section 2.2.4.7. Hence, usage of such kind of algorithms is recommended to obtain TIDs. Each dedicated TID is only used once. Moreover, linking of different ECDSA signatures generated by the same long term secret key to a common source node without knowledge of the corresponding long term public key, which is kept secret in the common VANET PKI approaches, is considered infeasible.

Fortunately, the unpredictability of a future TID does not only have a privacy benefit, but also provides a robustness increase for the operation of the CA. The CA performs a TID look-up for each received message, i.e., it is compared to the expected TIDs. In combination with a well performing storage for expected TIDs, e.g., in a hash table linking TIDs to corresponding public keys, such kind of look-up can be expected to be computationally cheap. In this process, the TID is used like an One Time Password (OTP).

In case the TID supplied by an attacker does not match any expected input, the request is dropped without performing any kind of computationally expensive cryptographic operation. Thus, the computational load generated by the attacker is significantly reduced in comparison to the standardized scheme. A more detailed comparison is given in the conducted evaluation of the approach.

To tolerate message loss, which also leads to valid TIDs never reaching a CA, multiple TIDs with future validity can be stored at the CA for each node. In case a received TID matches anyone out of the n_{TID} TIDs stored for each node, the message passes the basic input verification. Hence, the receiver proceeds with verification of the digital signature.

The presence of the plaintext TID allows to verify the digital signature of the request before decrypting it. Hence, even in case an attacker can obtain a valid TID, only one computationally demanding cryptographic operation is performed before the attack gets detected, which leads to the request being discarded.

6.4.2.1 Attack Possibilities on the TID Scheme

An attacker can try to misuse a monitored TID by sending a bogus request with this dedicated TID to the corresponding CA. This attack succeeds in case the bogus message arrives sooner at the CA than the original message (wormhole attack).

If an attacker can ban the request to arrive at the CA, the attack will clearly succeed. Thus, the attackers message's digital signature gets verified. However, each TID is only valid once. Hence, the computational load an attacker can cause at the CA is highly limited assuming that the attacker can only control the message flow of a limited number of nodes.

Another kind of attack is a DOS attack on nodes requesting PSCs from CAs. In case the attacker can perform the above outlined attack more than n_{TID} times in a row for one node, synchronization of the internal stati of the hash chains within node and CA is lost.

A possible countermeasure to these attacks is to enforce a minimum time interval between successive requests from the same node, an approach well known from password query systems.Thereby, the number of valid TIDs, which an attacker can obtain from a well controlled set of nodes, can be significantly limited. Moreover, there is a high chance that nodes move out of the attacked areas of local static attackers within the time interval between multiple successive requests.

However, global or mobile attackers, which can control the message flow of an attacked node over a longer time span can still perform the attack. However, such kind of attackers can also ban a node from communicating within the VANET by multiple other mechanisms, like dedicated jamming of messages from the attacked node. Hence, the TID mechanism is regarded as not extending the capabilities of such already (very) powerful attackers.

6.4.2.2 Extension Towards a Symmetric Key Signature Scheme

Instead of using an ECDSA signature, other more computationally efficient signature schemes could be used. A well known symmetric key approach is to obtain digital signatures from a Message Authentication Code (MAC, not to be confused with Medium Access Control (MAC)).

The approach of using a TID already requires a shared secret (i.e., secret key) known to node and CA. Thus, this shared secret can be used to apply concepts of symmetric key cryptography to the data retrieval procedure.

Many different approaches for obtaining signature keys from a shared secret have been proposed [223, 244]. Thereby, the TID usage scheme can combined with any standard message authentication code algorithm, e.g., hash based MAC (HMAC) [244], while keeping the encryption part unchanged to avoid to introduce too many extra concepts.

The approach based on symmetric keys has three more advantages over the design from [105], in comparison with the above described proposal still using ECDSA signatures. These are

1. increased robustness against DOS attacks at receivers, as verification of a symmetric key signature bears much less effort in comparison to asymmetric signature schemes,

2. lowered computational effort at the node sending a request, as creation of a symmetric key signature bears much less effort in comparison to asymmetric key schemes, and

3. reduced request message size, as symmetric key signatures achieve comparable security levels with reduced length in comparison to public key schemes [244].

A possible drawback of including the proposed symmetric key signature scheme into a VANET approach is increased system complexity in comparison to the public key only approach. Two different signature schemes have to be provided instead of only a single one.

6.4.2.3 Evaluation of Certificate Refill Request Schemes

To evaluate the proposed approaches, runtime measurements were conducted for each of them. The used implementation uses C++ and cryptographic primitives from the well known Crypto++ framework [75]. The evaluation platform is the Intel Core i7 described in Section 3.4. Moreover, all measurement results are obtained using the measurement methodology from Section 3.4.

Furthermore, THF (from [28]) is used to obtain TIDs. It uses SHA-512 for h_0 (internal step) and SHA-3 for h_1 (output step), i.e., generation of individual TIDs. The length of a TID is 128 bits. Mapping of TIDs to corresponding public keys is implemented via a hash map.

Three scenarios are regarded for evaluation. At first, messages holding valid signatures are used to resemble normal operation mode, i.e., no attack is present. Secondly, messages sent by an attacker without knowledge of any secret key material or valid TIDs is considered to mimic a DOS attack. At last, messages holding an invalid signature, but a valid TID, are used to show the impact of an advanced attack. The TIDs are either guessed correctly just by chance or they are obtained by an advanced attack, like the one described above.

Measured runtimes for the input verification of messages following the considered request message types are given in Figure 6.17. The standard deviations of measurement results shown

in Figure 6.17 are quite small. For all results the Coefficient of Variation (CV) is lower then 10^{-2}. Thus, the obtained results can be regarded as being reliable.

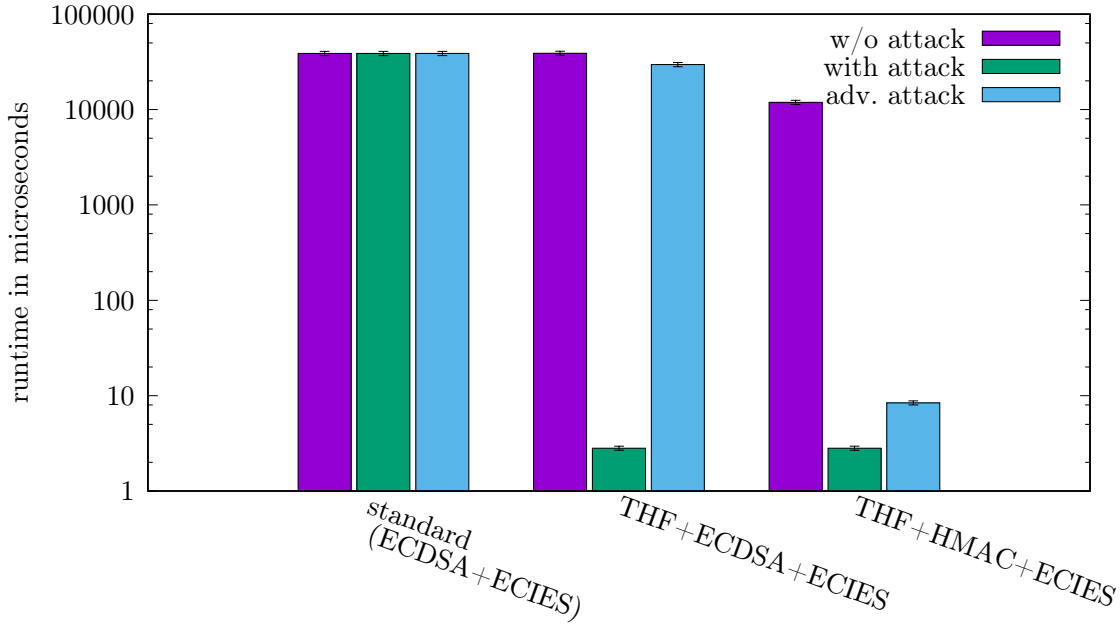


Figure 6.17: Runtime for verification of a PSC refill request at a CA.

Results from Figure 6.17 clearly show the design advantage of the approaches using TIDs over the standardized mechanism for the case of an attack. Especially, in case of the non-advanced attack both schemes can massively limit the computational load caused by a bogus refill request issued by an attacker. In case of the advanced attack, both schemes provide reduced computational effort at the CA by sparing decryption of the bogus message. However, the gain is clearly bigger in case the approach using a symmetric key signature (see Section 6.4.2.2) is applied. Moreover, the runtime for valid requests can be limited by the scheme using a symmetric key signature. In contrast, both other schemes show an almost identical runtime for processing valid PSC refill requests. These results show that the proposed mechanisms for avoiding a DOS weakness of the PSC refill process work well.

6.5 Cross Influence between Certificate Distribution and Pseudonym Change

Characteristics of the cross influence between PSC distribution algorithms and pseudonym change (i.e., PSC change) algorithms have been introduced in Section 4.4. Thereby, pseudonym change is identified as a possibly major source of superfluous PSC emissions. Such kind of extra PSC emissions can deteriorate the bandwidth saving effects intended by only sporadic PSC inclusion, as used by many popular PSC distribution strategies (see also Section 2.2.4.3). The

approach described in the following is also part of the author's prior work in [42][4].

To avoid the majority of superfluous PSC emissions, explicit signaling of a pseudonym change after the change itself can be used. To realize this, a node adds an extra flag to the security envelope of the first beacon message after the pseudonym change. All nodes receiving this message do not trigger detection of a new node within the security entity based on this message. I.e., neighborhood aware PSC emission is disabled for this dedicated message. Hence, superfluous PSC emissions are avoided.

Evaluation of the proposed signaling is done within the environment described in Chapter 3. The freeway scenario (see Section 3.2) is used in combination with two different pseudonym change methods. These are timeout based pseudonym change and a mix zone approach with temporarily disabled message sending within the mix zone [154, 294].

Obtained results for the uncoordinated pseudonym change method from current standards are given in Table 6.1. To obtain the given figures, a pseudonym change interval of 30 s is used within all nodes. A randomly chosen offset t for the pseudonym change timeout with $0\text{ s} < t < 30\text{ s}$ is used at each node when it gets inserted into the simulation to avoid unintended synchronization of pseudonym changes within the simulation environment.

scenario (node interval)	average CHBR in %	σ^2
without signaling (2 s)	55.0	1.67
with signaling (2 s)	40.5	1.79
without PSC change (2 s)	39.5	1.52
without signaling (9 s)	22.2	1.61
with signaling (9 s)	15.9	1.73
without PSC change (9 s)	15.7	1.71

Table 6.1: CHBR for uncoordinated pseudonym switching with and without explicit signaling of the pseudonym change.

The results from Table 6.1 clearly show that the used approach is able to significantly limit the experienced channel load. In scenarios with applied a-posteriori signaling of PSC changes the increase in comparison to the corresponding reference scenarios without a performed PSC change is quite small. A significant limitation in channel load in comparison to the standardized system without signaling is achieved.

Evaluation results for the mix zone approach are given in Figure 6.19. To obtain the provided figures, a rectangular shaped mix zone along the freeway scenario from Section 3.2 with a length of 50 m is used. It is illustrated in Figure 6.18.

The channel load increase caused by the PSC changes within the mix zone can be significantly limited with the proposed signaling approach in comparison to the standardized system, as shown in Figure 6.19. The improvement is more significant for the scenario with more dense traffic (average node interval 2 s) than in the more sparse scenario. This is due to the more significant negative impact of the superfluous new neighbor detections in the high density scenario.

[4] See also footnote 6.

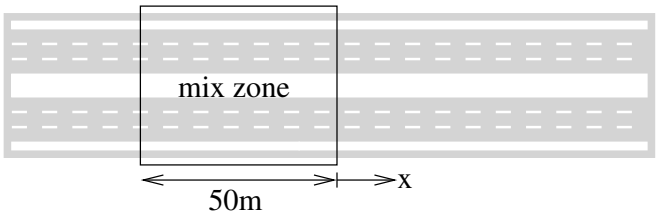


Figure 6.18: Sketch of mix zone along a freeway as used for evaluation.

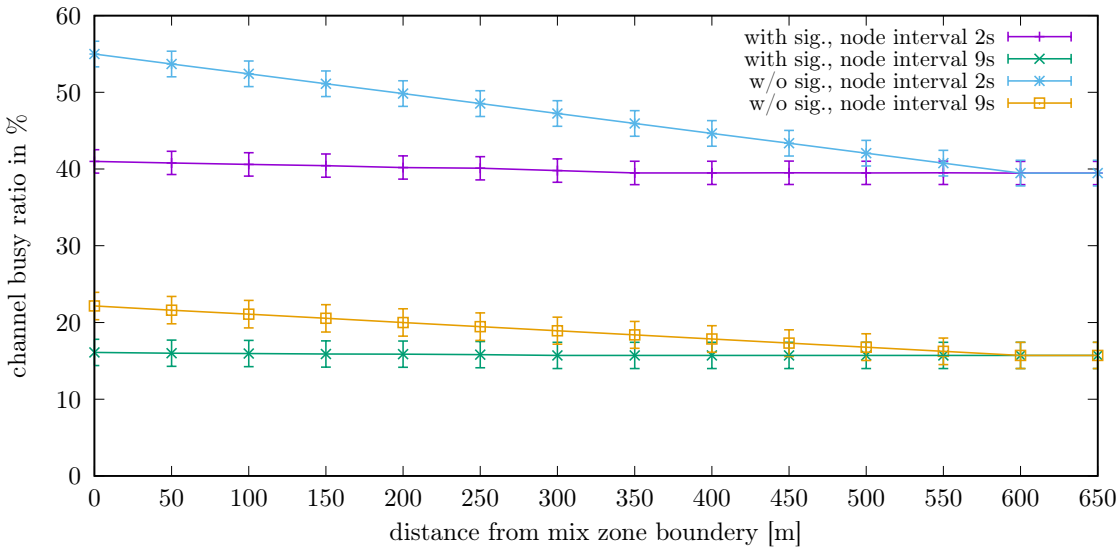


Figure 6.19: CHBR for mix zone based pseudonym change with and without explicit signaling of the changes.

In this case more nodes detect the “new” neighbors in comparison to the lower density scenario, i.e., with a node interval of 9 s.

Obtained results show that both uncoordinated and coordinated, i.e., mix zone based, pseudonym change can profit from explicit signaling of the change. Superfluous new neighbor detections can be avoided yielding less CHBR, which leads to improved communication conditions for all nodes in vicinity of the node(s) performing the pseudonym change.

The following section studies the feasibility of the current specification of encrypted multi-hop communication in ETSI ITS. This kind of confidential communication is required to request PSC updates, as stated before.

6.6 Encrypted Multi-Hop Communication

Encryption is required to provide confidentiality of communication in VANETs. One important application of encrypted communication in VANETs is PSC refill, as required to realize requirements no. 3 and 4 from Section 5.5.

However, an in-detail analysis of the current structuring of data from network layer and security entity within ETSI ITS shows that end-to-end encrypted multi-hop communication is not possible following current standards [122, 125]. This is caused by the fact that with the move of version 1.1.1 [109] to 1.2.1 [125] of the security envelope's standard the possibility of multiple payload fields within the security envelope got removed. This follows criticism of the multi-payload support in [237]. Thus, the routing information of a message is now handed over to the security entity together with the to be encrypted higher level data, as a common data block. Hence, the content of the security envelope is as illustrated in Figure 6.20.

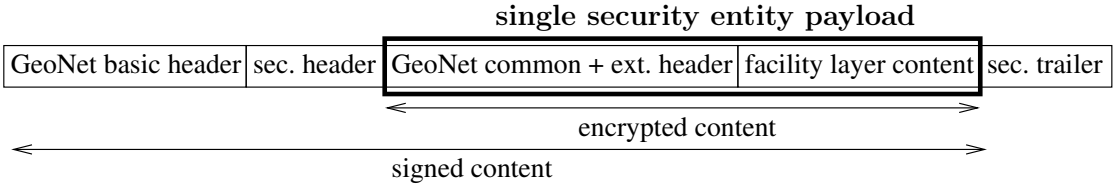


Figure 6.20: Standardized security envelope for an encrypted multi-hop message.

One can see from Figure 6.20, that the single payload block including the routing information required by forwarders gets encrypted together with higher level payload. Thus, forwarders cannot access it, as it is encrypted using credentials only known to the final target of the message. Hence, forwarders cannot obtain the required routing information and have to drop the message. This entirely disables end-to-end encrypted multi-hop communication within ETSI ITS.

Mainly two approaches can be thought of to overcome the found issue. One can either

1. move data encryption to a higher layer, i.e., the network layer level security functionality always realizes only signing, and never encryption, or

2. one (partly) re-enables support for multiple payloads. Thereby, the network layer uses signing for its meta data. Moreover, signing together with optional encryption can be used for the higher level payload.

Approach no. 1 requires to introduce a new interface between the facility layer and the security entity to provide data encryption. Moreover, a second kind of security envelope on the facility layer has to be defined, to hold the corresponding meta data required for data decryption. Thus, this would significantly increase the complexity of the overall protocol stack design. Hence, this approach is not recommended.

To enable proposal no. 2, one just needs to allow at most two differently treated payload fields within the security envelope. This needs only to be used in case of an encrypted multi-hop message. For all other message types a single, signed payload field can be used. However, to enable secure message routing, the routing information (first payload) needs to be signed together with the encrypted higher layer payload (second payload). This design is illustrated in Figure 6.21. It enables forwarders to access the required routing information, while keeping the confidential higher level data secret.

The proposed approach avoids the high complexity of full support for an arbitrary mixture of multiple payloads within the security envelope, which is criticized in [237], while enabling end-to-end encrypted multi-hop communication in ETSI ITS. Thus, its future usage is recommended.

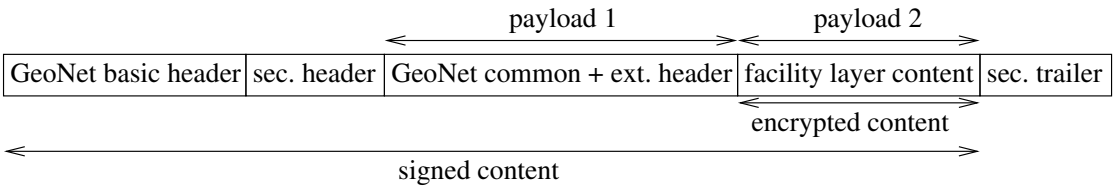


Figure 6.21: Usable security envelope for an encrypted multi-hop message.

Section 6.7 provides a summary about results obtained throughout individual sections of this chapter.

6.7 Summary of Proposals for Advanced Certificate Handling

Several proposals for improved certificate handling have been proposed in prior sections of this chapter. These are based on findings in regard to security related overhead from Chapter 4 and requirements emerging from advanced attacks discussed in Chapter 5. In summary, it is proposed to

1. use an adaptive, situation aware PSC distribution strategy to limit both channel load caused by PSC distribution and cryptographic packet loss,
2. replace full certificate chain distribution and corresponding requests by a combination of requests for dedicated certificates and only single certificate distribution to avoid high channel load causing also a vulnerability to DOS attacks,
3. introduce dedicated certificates for securing PSC update requests to enable such updates in scenarios without an available, valid PSC at the requester,
4. use an improved message format for PSC update requests utilizing one-time identifiers for increased DOS robustness of CAs,
5. introduce a mechanism for certificate change signaling to avoid unnecessary PSC emissions, due to superfluous new neighbor detections,
6. fix a design fault in the message format of encrypted multi-hop messages to re-enable forwarding of such messages.

For more details the reader is referred to corresponding prior sections of this chapter. The following chapter provides a conclusion about results obtained in this work alongside with possible topics of future work.

Chapter 7

Conclusions and Future Work

Vehicular ad-hoc networks (VANETs) are an important approach to increase future safety of driving by enabling cooperative Advanced Driver Assistence Systems (ADASs). However, rigid security and privacy requirements employed to conducted wireless data exchange still pose significant challenges for VANET approaches. Several weaknesses of the current state of the art of VANET approaches from ETSI ITS as well as WAVE standard frameworks have been identified in this work.

Three main attack surfaces of ETSI ITS and WAVE based VANETs are identified in this thesis, which are

1. constant and distinctive content in various data fields within frequently sent VANET messages highly endanger privacy of vehicles, and thereby also their drivers,

2. the distribution strategy of Certificate Authority (CA) certificates allows even a simple static outsider attacker to massively increase the channel load within a large area around the attacker, which significantly exceeds his own communication range, and

3. GNSS spoofing modifying time and position information inside nodes

   (a) endangers the basic system requirement of accountability by circumventing the non-repudiation feature of the employed digital signature scheme,

   (b) endangers the access control system by forcing the acceptance of outdated messages and certificates, and

   (c) enables an attacker to perform a Sybil attack.

Hence, the identified security problems need to be overcome to re-enable secure usage of VANETs and ADASs, which are based on the information obtained via VANETs.

Moreover, several communication protocol design weaknesses of the ETSI ITS approach have been identified. It is found that the standardized way of cross layer message assembly leads to frequent violation of low layers' maximum packet size restrictions. This causes inabilities to distribute important data sets from the application layer. Furthermore, confidential end-to-end encrypted communication over a multi-hop connection is impossible, as forwarders

cannot access required routing information. This is caused by incorrect data encryption rules. Mechanisms to overcome the found shortcomings are proposed.

To overcome the outlined security issues, several improvements to VANET mechanisms have been proposed. These include,

1. secure time synchronization between nodes, but state of the art mechanisms can hardly provide it,
2. presence of multiple pseudonym certificates being valid during the same time span within a node is to be avoided,
3. pre-caching of pseudonym certificates valid in the future within nodes should be limited to a minimum,
4. presence of constant but distinctive data sets within VANET messages has to be avoided to enable privacy conserving pseudonym changes,
5. mechanisms for limiting the channel load caused by certificate distribution are required, especially
   (a) after a pseudonym change the number of superfluous pseudonym certificate distributions due to new neighbor detection should be limited by using explicit signaling of the change,
   (b) sending of certificate chains should be removed altogether, instead individual dissemination should be used for CA certificates, and
   (c) the number of CA certificate deliveries after a request for such a kind of certificate should be limited to a minimum by using targeted requests.

By employing the given improvements most of the found security weaknesses can be overcome (issues 1, 2 and 3c). For the remaining weaknesses the required capabilities for a successful attack can be made significantly more challenging.

Additionally, an evaluation of security related data set sizes achieved by different platform independent data representation schemes is provided. Its results show that standardized usage of ASN.1 yields a significantly longer average message size in comparison to an EXI based scheme. Hence, EXI based data representation should be used for the data sets inside the ETSI ITS messages' security envelope.

Future work should define requirements on the accuracy of absolute location information inside vehicles. This should be followed by the development of appropriate solutions to obtain this data set in a secure manner. GNSS spoofing attacks show that pure satellite based positioning is currently not able to provide important location information in a secure manner. Hence, future work is required in this area.

Moreover, future work should study the robustness of non-GNSS based wireless time synchronization mechanisms against spoofing attacks in more detail. In doing so, it can be shown whether such mechanisms can provide a secure alternative for GNSS based time synchronization inside a VANET. In general, the question of how to provide accurate and secure time synchronization of highly mobile nodes with only sporadic backbone connections to the global reference

time defined by the backbone network is an open issue. Its solution is important not only for VANET protocol design. Instead, such a solution is required for all kinds of wireless ad-hoc networks, which rely on accurate time synchronization to realize their use cases in a secure manner. Hence, this topic should be addressed in future work.

Appendix A

Abbreviations

AA	Authorization Authority
AAC	Authorization Authority Certificate
ADAS	Advanced Driver Assistence System
AES	Advanced Encryption Standard
AID	Application Identifier
ASN.1	Abstract Syntax Notation 1
AU	Application Unit
BER	Basic Encoding Rules
BSM	Basic Safety Message
BTP	Basic Transport Protocol
C2C-CC	Car2Car Communication Consortium
C2X	Car-to-X
CA	Certificate Authority
CABS	Cooperative Awareness Basic Service
CAM	Cooperative Awareness Message
CAN	Controller Area Network
CBF	contention-based forwarding
CBR	contention-based responding
CCU	Communication and Control Unit
CDD	Common Data Dictionary
CDF	Cumulated Distribution Function
CHBR	channel busy ratio
CRC	Cyclic Redundancy Check
CRL	Certificate Revocation List
CSMA-CA	carrier sense multiple access - collision avoidance
CSR	Certificate Signing Request
CV	Coefficient of Variation
DCC	Decentralized Congestion Control
DENBS	Decentralized Environment Notification Basic Service
DENM	Decentralized Environment Notification Message

DER	Data Encoding Rules
DOS	Denial of Service
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
ECU	Electronic Control Unit
ETSI	European Telecommunications Standards Institute
EXI	Efficient XML Interchange
exip	Embeddable EXI Processor in C
FIFO	first in first out
GBC	Geo-Broadcast
GNSS	Global Navigation Satellite System
GCC	GNU Compiler Collection
protobuf	Google Protocol Buffers
GPS	Global Positioning System
gpsd	GPS deamon
HMAC	hash based MAC
HMI	human machine interface
HOTP	HMAC-based One-time Password Algorithm
HSM	Hardware Security Module
ITS	Intelligent Transport Systems
JSON	JavaScript Object Notation
LDM	Local Dynamic Map
LLC	Logical Link Control
LTC	Long Term Certificate
MAC	Medium Access Control
MANET	Mobile ad-hoc network
NITZ	Network Identity and Time Zone
NTP	Network Time Protocol
ntpd	NTP deamon
OBU	on-board unit
OEM	Original Equipment Manufacturer
OTP	One Time Password
PCA	Pseudonym Certificate Authority
PER	Packed Encoding Rules
PKI	Public Key Infrastructure
PSC	Pseudonym Certificate
RSU	road side unit
SDN	Software Defined Networking
SHA	Secure Hash Algorithm
SHB	Single Hop Broadcast
SSP	Service Specific Permission
SUMO	Simulation of Urban MObility

SVG	Scalable Vector Graphics
TID	Temporary Identifier
TLS	Transport Layer Security
TOTP	Time-based One-time Password Algorithm
TSC	time stamp counter
UDP	User Datagram Protocol
UPER	Unaligned Packed Encoding Rules
USRP	Universal Software Radio Peripheral
UTC	Coordinated Universal Time
V2X	Vehicle-to-X
VANET	Vehicular ad-hoc network
VoD	Verify-on-Demand
VPN	Virtual Private Network
WAVE	Wireless Access in Vehicular Environments
WSME	WAVE Station Management Entity
WSMP	WAVE Short Message Protocol
WSN	Wireless Sensor Network
XML	Extensible Markup Language

Bibliography

[1] "Memorandum of Understanding for OEMs within the CAR 2 CAR Communication Consortium on Deployment Strategy for cooperative ITS in Europe," June 2011, v 4.0102.

[2] "ntpd - Network Time Protocol (NTP) Daemon," online: http://www.eecis.udel.edu/~mills/ntp/html/ntpd.html, Mar. 2014, retrieved: 03.2017.

[3] "gpsd - a GPS service daemon," online: http://catb.org/gpsd/index.html, May 2016, retrieved: 03.2017.

[4] "Safeguarding of Data Exchange," European Patent EP 3 133 769 A1, Feb., 2017.

[5] M. Adalier, "Efficient and Secure Elliptic Curve Cryptography Implementation of Curve P-256," in *Workshop on Elliptic Curve Cryptography Standards*, June 2015.

[6] A. Adigun, B. A. Bensaber, and I. Biskri, "Protocol of Change Pseudonyms for VANETs," in *9th IEEE International Workshop on Performance and Management of Wireless and Mobile Networks*, Oct. 2013, pp. 162–167.

[7] S. Al-Kuwari and D. Wolthusen, "On the Feasibility of Carrying Out Live Real-Time Forensics for Modern Intelligent Vehicles," in *Forensics in Telecommunications, Information and Multimedia: Third International ICST Conference*, 2010, pp. 207– 223.

[8] N. Alam, A. T. Balaei, and A. G. Dempster, "Relative Positioning Enhancement in VANETs: A Tight Integration Approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 47–55, July 2012.

[9] *AMD Geode LX Processor Family*, AMD, Feb. 2014, doc. No. 33358E.

[10] S. Ammoun and F. Nashashibi, "Real Time Trajectory Prediction for Collision Risk Estimation between Vehicles," in *IEEE 5th International Conference on Intelligent Computer Communication and Processing*, Aug. 2009, pp. 417–422.

[11] I. I. Androulidakis, *Mobile Phone Security and Forensics: A Practical Approach*. Springer, Mar. 2016, vol. 2nd.

[12] D. Angermeier, A. Kiening, and F. Stumpf, "PAL - Privacy Augmented LTE: A Privacy-Preserving Scheme for Vehicular LTE Communication," in *Proceeding of the Tenth ACM*

International Workshop on Vehicular Inter-Networking, Systems, and Applications, June 2013, pp. 1–10.

[13] P. G. Argyroudis, R. Verma, H. Tewari, and D. O'Mahony, "Performance Analysis of Cryptographic Protocols on Handheld Devices," in *Third IEEE International Symposium on Network Computing and Applications*, Aug. 2014, pp. 169–174.

[14] A. A. Atayero and Y. A. Ivanov, *Integrated Models for Information Communication Systems and Networks: Design and Development*. IGI Global, June 2013, ch. Modeling of Packet Streaming Services in Information Communication Networks, pp. 166–206.

[15] Atmel Corporation, "AVR411: Secure Rolling Code Algorithm for Wireless Link," Atmel Corporation, Tech. Rep. 2600E-AVR-07/15, July 2015.

[16] M. Aydos, B. Sunar, and C. K. Koc, "An Elliptic Curve Cryptography based Authentication and Key Agreement Protocol for Wireless Communication," in *2nd International Workshoop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Oct. 1998, pp. 1–12.

[17] S. Bai, "US - EU V2V V2I Message Set Standards Collaboration," online: https://docbox.etsi.org/workshop/2014/201402_ITSWORKSHOP/S02_ITS_SomeBitsFromtheWorld/HONDA_BAI.pdf, Feb. 2013, ETSI ITS Workshop, retrieved: 03.2017.

[18] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO - Simulation of Urban MObility: An Overview," in *The Third International Conference on Advances in System Simulation*, Oct. 2011, pp. 63–68.

[19] F. Bellard, "FFASN1 Compiler," online: http://bellard.org/ffasn1/, Sept. 2012, retrieved: 03.2017.

[20] A. Bensky, *Wireless Positioning Technologies and Applications*, 1st ed., ser. GNSS Technology and Applications Series. Artech House, 2007.

[21] K. Bicakci and N. Baykal, "Infinite Length Hash Chains and their Applications," in *Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, June 2002, pp. 57–61.

[22] T. Bingmann, "malloc_count - Tools for Runtime Memory Usage Analysis and Profiling," online: http://panthema.net/2013/malloc_count/, Sept. 2014, retrieved: 03.2017.

[23] N. Bißmayer, "Misbehavior Detection and Attacker Identification in Vehicular Ad hoc Networks," Ph.D. dissertation, Technische Universität Darmstadt, Oct. 2014.

[24] N. Bißmeyer, H. Stubing, E. Schoch, S. Gotz, J. P. Stotz, and B. B. Lonc, "A Generic Public Key Infrastructure for Securing Car-to-X Communication," in *18th ITS World Congress*, Dec. 2011.

[25] N. Bißmeyer, H. Stübing, E. Schoch, S. Götz, J. P. Stotz, and B. Lonc, "A generic public key infrastructure for securing car-to-x communication," in *World Congress on Intelligent Transport Systems*, Oct. 2011.

[26] S. Bittl, "Angriffspotentiale und Effiziente Absicherung Automobiler Bussysteme als Grundlage sicherer Car2X-Kommunikation," in *11th Conference Wireless Communication and Information*. vwh, Oct. 2014, pp. 37–49.

[27] ——, "Attack Potential and Efficient Security Enhancement of Automotive Bus Networks using Short MACs with Rapid Key Change," in *6th International Workshop Nets4Cars/Nets4Trains/Nets4Aircraft 2014*, ser. LNCS, vol. 8435. Springer, May 2014, pp. 113–125.

[28] ——, "Efficient Construction of Infinite Length Hash Chains with Perfect Forward Secrecy using Two Independent Hash Functions," in *11th International Conference on Security and Cryptography*. SCITEPRESS Digital Library, Aug. 2014, pp. 213 – 220.

[29] ——, "Towards Solutions for Current Security Related Issues in ETSI ITS," in *10th International Workshop Nets4Cars/Nets4Trains/Nets4Aircraft*, ser. LNCS, vol. 9669. Springer, May 2016, pp. 136 – 148.

[30] ——, "Privacy Conserving Low Volume Information Retrieval from Backbone Services in VANETs," *Vehicular Communications*, vol. 9C, pp. 1–7, Feb. 2017.

[31] S. Bittl, B. Aydinli, and K. Roscher, "Distribution of Pseudonym Certificates via Bursts for VANETs with Low and Medium Mobility," in *8th IFIP Wireless Mobile Networking Conference*, Oct. 2015, pp. 227 – 230.

[32] ——, "Effective Certificate Distribution in ETSI ITS VANETs using Implicit and Explicit Requests," in *8th International Workshop Nets4Cars/Nets4Trains/Nets4Aircraft*, ser. LNCS, M. Kassab et al., Ed., vol. 9066. Springer, May 2015, pp. 72–83.

[33] ——, "Efficient Rate-Adaptive Certificate Distribution in VANETs," in *Twelfth International Symposium on Wireless Communication Systems*. SCITEPRESS Digital Library, Aug. 2015, pp. 371 – 375.

[34] S. Bittl and A. A. Gonzalez, "Privacy Issues and Pitfalls in VANET Standards," in *1st International Conference on Vehicular Intelligent Transport Systems*. SCITEPRESS Digital Library, May 2015, pp. 144 – 151.

[35] ——, *Smart Cities, Green Technologies, and Intelligent Transport Systems*, ser. CCIS. Springer, Jan. 2016, vol. 579, ch. Privacy Endangerment from Protocol Data Sets in VANETs and Countermeasures, pp. 304–321.

[36] S. Bittl, A. A. Gonzalez, and W. Heidrich, "Performance Comparision of Encoding Schemes for ETSI ITS C2X Communication Systems," in *Third International Conference on Advances in Vehicular Systems, Technologies and Applications*. ThinkMind(TM) Digital Library, June 2014, pp. 58–63.

[37] S. Bittl, A. A. Gonzalez, M. Myrtus, H. Beckmann, S. Sailer, and B. Eissfeller, "Emerging Attacks on VANET Security based on GPS Time Spoofing," in *IEEE Communications and Network Security Conference*, Sept. 2015, pp. 344 – 352.

[38] S. Bittl, A. A. Gonzalez, M. Spähn, and W. Heidrich, "Performance Comparison of Data Serialization Schemes for ETSI ITS Car-to-X Communication Systems," *International Journal On Advances in Telecommunications*, vol. 8, pp. 48 – 58, June 2015.

[39] S. Bittl and K. Roscher, "Efficient Authorization Authority Certificate Distribution in VANETs," in *2nd International Conference on Information Systems Security and Privacy*. SCITEPRESS Digital Library, Feb. 2016, pp. 85–96.

[40] ——, "Feasibility of Verify-on-Demand in VANETs," in *4th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication*, Apr. 2016, pp. 10 – 13.

[41] ——, *Information Systems Security and Privacy*, ser. CCIS. Springer, Feb. 2017, vol. 691, ch. Efficient Distribution of Certificate Chains in VANETs, pp. 86–107.

[42] ——, "Mutual Influence of Certificate Distribution and Pseudonym Change Strategies in Vehicular ad-hoc Networks," *International Journal of Vehicle Information and Communication Systems*, vol. 3, no. 2, pp. 158–172, Sept. 2017.

[43] ——, "Protocol Modeling Accuracy in VANET Simulators," in *5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication*, Apr. 2017, pp. 13 – 16.

[44] S. Bittl, K. Roscher, and A. A. Gonzalez, "Security Overhead and its Impact in VANETs," in *8th IFIP Wireless Mobile Networking Conference*, Oct. 2015, pp. 192 – 199.

[45] S. Bittl, M. Schlegel, and K. Roscher, "Simulationsbasierte Evaluierung eines zeit- und ortsbasierten Pseudonym-Wechsel-Verfahrens für ETSI ITS - Dezentraler Ansatz zur Verbesserung der Privatsphäre von Fahrern," in *31. VDI/VW-Gemeinschaftstagung Automotive Security*, ser. VDI-Berichte, vol. 2263. VDI Verlag, Oct. 2015, pp. 137 – 146.

[46] S. Bittl, D. Seydel, J. Pfeiffer, and J. Jiru, *SMARTGREENS/VEHTIS 2016 - Revised Selected Papers*, ser. CCIS. Springer, ch. Evaluation Methodology for Cooperative ADAS utilizing Simulation and Experiments, to appear.

[47] B. Blum, T. He, and S. Son, "IGF: A State-Free Robust Communication Protocol for Wireless Sensor Networks," Department of Computer Science, University of Virginia, Tech. Rep. CS-2003-11, 2003.

[48] M. Boban, "Realistic and Efficient Channel Modeling for Vehicular Networks," PhD thesis, Dept. of Electrical and Computer Engineering, Carnegie Mellon University, Dec. 2012.

[49] S. Born, "How to manipulate a radio controlled clock via speaker," online: http://bastianborn.de/radio-clock-hack, May 2014, retrieved: 03.2017.

[50] M. S. Bouassida and M. Shawky, "A Cooperative and Fully-distributed Congestion Control Approach within VANETs," in *9th International Conference on Intelligent Transport Systems Telecommunications*, Oct. 2009, pp. 526–531.

[51] C. Bournez, "Efficient XML Interchange Evaluation," W3C, Tech. Rep., Apr. 2009.

[52] J. Breu, A. Brakemeier, and M. Menth, "A quantitative study of Cooperative Awareness Messages in production VANETs," *EURASIP Journal on Wireless Communications and Networking*, vol. 98, pp. 1–18, June 2014.

[53] D. R. L. Brown, R. P. Gallant, and S. A. Vanstone, "Provably Secure Implicit Certificate Schemes," in *5th International Conference on Financial Cryptography*, Feb. 2002, pp. 156–165.

[54] Buburuzan, T. et al., "Draft C2C-CC Standards System Profile," CAR 2 CAR Communication Consortium, Tech. Rep., Jan. 2014, v1.0.4.

[55] Y. Bulygin, "ASN.1 parsing in crypto libraries: what could go wrong?" Fourth International Conference on Cryptology and Information Security in Latin America, Invited Talk, Aug. 2015.

[56] S. Busanelli, G. Ferrari, and V. A. Giorgio, "I2V Highway and Urban Vehicular Networks: A Comparative Analysis of the Impact of Mobility on Broadcast Data Dissemination," *Journal of Communications*, vol. 6, no. 1, pp. 87–100, Feb. 2011.

[57] C. Büttner and S. A. Huss, "An Anonymous Geocast Scheme for ITS Applications," in *2nd International Conference on Information Systems Security and Privacy*, Feb. 2016, pp. 179–189.

[58] L. Buttyan, T. Holczer, A. Weimerskirch, and W. Whyte, "SLOW: A Practical pseudonym changing scheme for location privacy in VANETs," in *IEEE Vehicular Networking Conference*, Oct. 2009, pp. 1–8.

[59] C. Campolo, A. Molinaro, and R. Scopigno, Eds., *Vehicular ad hoc Networks - Standards, Solutions, and Research*. Springer, Dec. 2015.

[60] *C2C-CC Basic System Standards Profile*, CAR 2 CAR Communication Consortium Std. 000 042, Rev. 1.0.5, Jan. 2014.

[61] S. Chakraborty and S. Ramesh, "Special Section on Automotive Embedded Systems and Software," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 11, pp. 1701–1703, Nov. 2015.

[62] Y. Chang, C. P. Lee, and J. A. Copeland, "Goodput Enhancement of VANETs in Noisy CSMA/CA Channels," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 236 – 249, Jan. 2011.

[63] B. K. Chausrasia and S. Verma, "Optimizing Pseudonym Updation for Anonymity in VANETS," in *IEEE Asia-Pacific Services Computing Conference*, Dec. 2008, pp. 1633–1637.

[64] L. Cheng, B. E. Henty, F. Bai, and D. D. Stancil, "Highway and Rural Propagation Channel Modeling for Vehicle-to-Vehicle Communications at 5.9 GHz," in *IEEE Antennas and Propagation Society International Symposium*, Jul. 2008, pp. 1–4.

[65] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, and P. Mudalige, "Mobile Vehicle-to-Vehicle Narrow-Band Channel Measurement and Characterization of the 5.9 GHz Dedicated Short Range Communication (DSRC) Frquency Band," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1501–1516, Oct. 2007.

[66] M. P. Clark, *Data Networks, IP and the Internet: Protocols, Design and Operation*. Wiley, Mar. 2003.

[67] Cohda Wireless, "MK4a V2X Evaluation Kit," online: http://cohdawireless.com/Portals/0/PDFs/CohdaWirelessMK4a.pdf, Jun. 2013, retrieved: 03.2017.

[68] ——, "Cohda Wireless - Hardware," online: http://cohdawireless.com/Products/Hardware.aspx, 2015, retrieved: 03.2017.

[69] D. Comer and D. L. Stevens, *Internetworking With Tcp/Ip: Principles, Protocols, and Architecture*, 1st ed. Prentice Hall, Mar. 1995, vol. 1.

[70] N. Cottin, "ASN.1 security issues," online: http://powerasn.ncottin.net/download/ASN1_SecurityIssues.pdf, Oct. 2007, retrieved: 03.2017.

[71] G. Coulouris, J. Dollimore, T. Kindberg, and G. Blair, *Distributed Systems, Concepts and Design*, 5th ed. Addison-Wesley, July 2011.

[72] J. Cowley, *Communications and Networking*, 2nd ed. Springer, Sept. 2012.

[73] R. Curnow, "User guide for the chrony suite," online: http://chrony.tuxfamily.org/manual.html, Apr. 2016, retrieved: 03.2017.

[74] B. Cusack and A. Nisbet, "Secure Key Deployment and Exchange Protocol for MANET Information Management," in *10th Australian Digital Forensics Conference*, Dec. 2012, pp. 1–9.

[75] W. Dai, "Crypto++ Library," online: http://www.cryptopp.com/, Nov. 2015, retrieved: 03.2017.

[76] DCAITI/TU Berlin, "VSimRTI - Smart Mobility Simulation," online: https://www.dcaiti.tu-berlin.de/research/simulation/, Dec. 2015, retrieved: 03.2017.

[77] L. Delgrossi and T. Zhang, *Vehicle Safety Communications: Protocols, Security, and Privacy*. Wiley, Nov. 2012.

[78] H. Deng, A. Mukherjee, and D. P. Agrawal, "Threshold and Identity-based Key Management and Authentication for Wireless Ad Hoc Networks," in *International Conference on Information Technology: Coding and Computing*, Apr. 2004, pp. 107–111.

[79] R. Di Pietro, A. Durante, L. Mancini, and V. Patil, "Practically Unbounded One-Way Chains for Authentication with Backward Secrecy," in *First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, Sep. 2005, pp. 400–402.

[80] R. Di Pietro, S. Guarino, N. V. Verde, and J. Domingo-Ferrer, "Security in Wireless Ad-Hoc Networks - A Survey," *Computer Communications*, vol. 51, pp. 1–20, Sept. 2014.

[81] R. Di Pietro, L. V. Mancini, A. Durante, and V. Patil, "Addressing the Shortcomings of one-way Chains," in *Proceedings of the ACM Symposium on Information, Computer and Communications Security*, Mar. 2006, pp. 289–296.

[82] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," IETF, Tech. Rep. RFC:5246, Aug. 2008.

[83] H. Dok, H. Fu, R. Echevarria, and H. Weerasinghe, "Privacy Issues of Vehicular Ad-Hoc Networks," *International Journal of Future Generation Communication and Networking*, vol. 3, no. 1, pp. 17–32, Mar. 2010.

[84] A. Dorri, S. R. Kamel, and E. Kheyrkhah, "Security Challenges in Mobile Ad Hoc Networks: A Survey," *International Journal of Computer Science & Engineering Survey*, vol. 6, no. 1, pp. 15–29, Feb. 2015.

[85] F. Dötzer, *5th International Workshop Privacy Enhancing Technologies: Revised Selected Papers*. Springer, June 2006, ch. Privacy Issues in Vehicular Ad Hoc Networks, pp. 197–209.

[86] J. Douceur, "The Sybil Attack," in *First International Workshop on Peer to Peer (P2P) System WISTP*, ser. LNCS, no. 5019, May 2008, pp. 106–116.

[87] F. Dressler, H. Hartenstein, O. Altintas, and O. K. Tonguz, "Inter-Vehicle Communication - Quo Vadis," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 170–177, June 2014.

[88] F. Dressler, C. Sommer, D. Eckhoff, and O. K. Tonguz, "Toward Realistic Simulation of Intervehicle Communication: Model, Techniques and Pitfalls," *IEEE Vehicular Technology Magazine*, vol. 6, no. 3, pp. 43–51, Sept. 2011.

[89] F. Dressler, C. Sommer, T. Gansen, and L. Wischhof, "Requirements and Objectives for Secure Traffic Information Systems," in *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, Sept. 2008, pp. 808–814.

[90] O. Dubuisson, *ASN.1 - Communication Between Heterogeneous Systems*. OSS Nokalva, June 2000.

[91] P. Eckersley, "How Unique Is Your Web Browser?" in *10th International Symposium Privacy Enhancing Technologies*, ser. LNCS, M. J. Atallah and N. J. Hopper, Eds., vol. 6205. Springer, July 2010, pp. 1–18.

[92] D. Eckhoff, "Simulation of Privacy-Enhancing Technologies in Vehicular Ad-Hoc Networks," Dissertation, Friedrich-Alexander Universität Erlangen-Nürnberg, Faculty of Engineering, Mar. 2016.

[93] D. Eckhoff, N. Sofra, and R. German, "A Performance Study of Cooperative Awareness in ETSI ITS G5 and IEEE WAVE," in *10th Annual Conference on Wireless On-demand Network Systems and Services*, Mar. 2013, pp. 196 – 200.

[94] D. Eckhoff, C. Sommer, T. Gansen, R. German, and F. Dressler, "Strong and Affordable Location Privacy in VANETs: Identity Diffusion Using Time-Slots and Swapping," in *IEEE Vehicular Networking Conference*, Dec. 2010, pp. 174–181.

[95] *The JSON Data Interchange Format*, ECMA International Standard ECMA - 404, Rev. 1.0, Oct. 2013.

[96] S. Eichler, "Strategies for Pseudonym Changes in Vehicular Ad Hoc Networks depending on Node Mobility," in *IEEE Intelligent Vehicles Symposium*, June 2007, pp. 541–546.

[97] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. T. M. Shalmani, "On the Power of Power Analysis in the Real World: A Complete Break of the KeeLoq Code Hopping Scheme," in *Advances in Cryptology - CRYPTO*, ser. LNCS, D. Wagner, Ed., vol. 5157, Aug. 2008, pp. 203 – 220.

[98] A. A. Eltahir, R. A. Saeed, and R. A. Mokhtar, "Vehicular Communication and Cellular Network Integration: Gateway Selection Perspective," in *IEEE Conference on Computer and Communications Engineering*, Sept. 2014, pp. 64–67.

[99] D. Engeler, "Performance Analysis and Receiver Architectures of DCF77 Radio-controlled Clocks," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 59, no. 5, pp. 869 – 884, May 2012.

[100] R. G. Engoulou, M. Bellaiche, S. Pierre, and A. Quintero, "VANET Security Surveys," *Computer Communications*, vol. 44, pp. 1–13, May 2014.

[101] *Intelligent Transport Systems (ITS); Communications Architecture*, ETSI European Standard 302 665, Rev. V1.1.1, Sept. 2010.

[102] *Intelligent Transport Systems (ITS); Security; Security Services and Architecture*, ETSI Technical Specification 102 731, Rev. V1.1.1, Sept. 2010.

[103] *Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer Part*, ETSI Technical Specification 102 687, Rev. V1.1.1, July 2011.

[104] *Intelligent Transport Systems (ITS); Security; Stage 3 mapping for IEEE 1609.2*, ETSI Technical Specification 102 867, Rev. V1.1.1, June 2012.

[105] *Intelligent Transport Systems (ITS); Security; Trust and Privacy Management*, ETSI Technical Specification 102 941, Rev. V1.1.1, June 2012.

[106] *Intelligent Transport Systems (ITS); STDMA recommended parameters and settings for cooperative ITS; Access Layer Part*, ETSI Technical Recommendation 102 861, Rev. V1.1.1, Jan. 2012.

[107] ETSI, "3rd ITS Cooperative Mobility Services Plugtests," online: http://www.etsi.org/news-events/events/665-plugtests-2013-itscms3, Nov. 2013, retrieved: 03.2017.

[108] *Intelligent Transport Systems (ITS); Facilities layer function; Facility Position and time management*, ETSI Technical Specification 102 890-3, Rev. V0.0.2, Jan. 2013.

[109] *Intelligent Transport Systems (ITS); Security; Security header and certificate formats*, ETSI Technical Specification 103 097, Rev. V1.1.1, Apr. 2013.

[110] *Intelligent Transport Systems (ITS); Security; Security header and certificate formats*, ETSI Technical Specification 103 097, Rev. V2.1.1, June 2013, draft.

[111] *Intelligent Transport Systems (ITS); V2X Applications; Part 1: Road Hazard Signalling (RHS) application requirements specification*, ETSI Technical Specification 101 539-1, Rev. V1.1.1, Aug. 2013.

[112] *Intelligent Transport Systems (ITS); V2X Applications; Part 3: Longitudinal Collision Risk Warning (LCRW) application requirements specification*, ETSI Technical Specification 101 539-3, Rev. V1.1.1, Nov. 2013.

[113] *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Network Identity and TimeZone (NITZ); Service description; Stage 1 (3GPP TS 22.042 version 12.0.0 Release 12)*, ETSI Technical Specification 122 042, Rev. V12.0.0, Oct. 2014.

[114] *Intelligent Transport Systems (ITS); OSI cross-layer topics; Part 7: Interface between security entity and access layer*, ETSI Technical Specification 102 723-7, Rev. V1.0.4, Aug. 2014.

[115] *Intelligent Transport Systems (ITS); OSI cross-layer topics; Part 7: Interface between security entity and facilities layer*, ETSI Technical Specification 102 723-9, Rev. V1.0.4, Aug. 2014.

[116] *Intelligent Transport Systems (ITS); OSI cross-layer topics; Part 8: Interface between security entity and network and transport layer*, ETSI Technical Specification 102 723-8, Rev. V1.0.4, Aug. 2014.

[117] *Intelligent Transport Systems (ITS); Users and applications requirements; Part 2: Applications and facilities layer common data dictionary*, ETSI Technical Specification 102 894-2, Rev. V1.2.1, Sept. 2014.

[118] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)*, ETSI European Norm 302 895, Rev. V1.1.1, Sept. 2014.

[119] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, ETSI European Standard 302 637-2, Rev. V1.3.2, Nov. 2014.

[120] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service*, ETSI European Standard 302 637-3, Rev. V1.2.1, Sept. 2014.

[121] *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 3: Network Architecture*, ETSI European Norm 302 636-3, Rev. V1.1.2, Mar. 2014.

[122] *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical Addressing and Forwarding for Point-to-Point and Point-to-Multipoint Communications; Sub-part 1: Media-Independent Functionality*, ETSI European Standard 302 636-4-1, Rev. V1.2.1, July 2014.

[123] *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 6: Internet Integration; Sub-part 1: Transmission of IPv6 Packets over GeoNetworking Protocols*, ETSI European Norm 302 636-6-1, Rev. 1.2.1, May 2014.

[124] *Intelligent Transport Systems (ITS); Cross Layer DCC Management Entity for operation in the ITS G5A and ITS G5B medium*, ETSI Technical Specification 103 175, Rev. 1.1.1, June 2015.

[125] *Intelligent Transport Systems (ITS); Security; Security header and certificate formats*, ETSI Technical Specification 103 097, Rev. V1.2.1, June 2015.

[126] *Intelligent Transport Systems (ITS); Radiocommunications equipment operating in the 5 855 MHz to 5 925 MHz frequency band; Harmonised Standard covering the essential requirements of article 3.2 of Directive 2014/53/EU*, ETSI European Norm 302 571, Rev. V2.0.0, Mar. 2016.

[127] EU-US ITS Task Force, Standards Harmonization Working Group, Harmonization Task Group 6, "Harmonized security policies for cooperative Intelligent Transport Systems create international benefits," https://ec.europa.eu/digital-single-market/news/harmonized-security-policies-cooperative-intelligent-transport-systems-create-international, Oct. 2015, retrieved: 03.2017.

[128] EU-US ITS Task Force, Standards Harmonization Working Group, Harmonization Task Groups 1 & 3, "Progress and Findings in the Harmonisation of EU-US Security and Communications Standards in the field of Cooperative Systems: EU-US Task Force - Reports from HTG1 and HTG3," https://ec.europa.eu/digital-single-market/en/news/progress-and-findings-harmonisation-eu-us-security-and-communications-standards-field, Apr. 2013, retrieved: 03.2017.

[129] European Commission, "Intelligent transport systems - Connected and automated driving (C-ITS)," online: http://ec.europa.eu/transport/themes/its/c-its_en.htm, Jan. 2016.

[130] ——, "Road Safety," online: http://ec.europa.eu/transport/road_safety/index_en.htm, Feb. 2016.

[131] B. S. Everitt and A. Skrondal, *The Cambridge Dictionary of Statistics.*, 4th ed. Cambridge University Press, Oct. 2010.

[132] R. Exel, T. Sauter, P. Ferrari, and S. Rinaldi, *Industrial Communication Technology Handbook*, 2nd ed. CRC Press, 2015, ch. Fault-Tolerant Clock Synchronization in Industrial Automation Networks.

[133] M. Feiri, J. Petit, and F. Kargl, "Evaluation of Congestion-based Certificate Omission in VANETs," in *IEEE Vehicular Networking Conference*, Nov. 2012, pp. 101 – 108.

[134] ——, "Efficient and Secure Storage of Private Keys for Pseudonymous Vehicular Communication," in *Proceedings of the 20th ACM workshop on Security, privacy & dependability for cyber vehicles*, Nov. 2013, pp. 9 – 18.

[135] M. Feiri, J. Petit, R. Schmidt, and F. Kargl, "The Impact of Security on Cooperative Awareness in VANET," in *IEEE Vehicular Networking Conference*, Dec. 2013, pp. 127 – 134.

[136] M. Fellendorf and P. Vortisch, "Microscopic traffic flow simulator VISSIM," in *Fundamentals of Traffic Simulation*. Springer New York, 2010, ch. 2, pp. 63–93.

[137] R. Fernandes, F. Vieira, and M. Ferreira, "VNS: An Integrated Framework for Vehicular Networks Simulation," in *IEEE Vehicular Networking Conference*, Nov. 2012, pp. 195 – 202.

[138] A. Festag, "Cooperative Intelligent Transport Systems Standards in Europe," in *IEEE Communications Magazine*, vol. 52, no. 12, Dec. 2014, pp. 166–172.

[139] A. Festag, L. Le, and M. Goleva, "Field Operations Tests for Cooperative Systems: A Tussle Between Research, Standard and Deployment," in *Proceedings of the Eighth ACM International Workshop on Vehicular Inter-Networking*, Sept. 2011, pp. 73 – 78.

[140] H. Füßler, H. Hartenstein, M. Martin, W. Effelsberg, and J. Widmer, "Contention-Based Forwarding for Street Scenarios," in *1st International Workshop in Intelligent Transportation*, Mar. 2004, pp. 155–160.

[141] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, "Contention-Based Forwarding for Mobile Ad Hoc Networks," *Elsevier's Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, Nov. 2003.

[142] J.-I. Gailly, "The gzip homepage," online: http://www.gzip.org, July 2003, retrieved: 03.2017.

[143] M. Galpin, "Using Internet data in Android Applications," online: http://www.ibm.com/developerworks/xml/library/x-dataAndroid/x-dataAndroid-pdf.pdf, June 2010, retrieved: 03.2017.

[144] P. Ganesan, R. Venugopalan, P. Peddabachagari, A. Dean, F. Mueller, and M. Sichitiu, "Modeling Encryption Overhead for Sensor Network Nodes," in *2nd ACM International Conference on Wireless Sensor Networks and Applications*, Sept. 2003, pp. 151–159.

[145] G. Gaubatz, J.-P. Kaps, E. Ozturk, and B. Sunar, "State of the Art in Ultra-low Power Public Key Cryptography for Wireless Sensor Networks," in *Third IEEE International Conference on Pervasive Computing and Communications Workshops*, Mar. 2005, pp. 146–150.

[146] M. Gerlach and F. Güttler, "Privacy in VANETs using Changing Pseudonyms - Ideal and Real," in *65th IEEE Vehicular Technology Conference*, Apr. 2007, pp. 2521–2525.

[147] Geschäftsstelle Verkehrsministerkonferenz, "Beschluss-Sammlung der Verkehrsministerkonferenz am 6./7. Oktober 2016 in Stuttgart," online: http://www.verkehrsministerkonferenz.de/VMK/DE/termine/sitzungen/16-10-06-07.html?nn=4812620, Oct. 2016, retrieved: 03.2017.

[148] B. Gil and P. Trezentos, "Impacts of data interchange formats on energy consumption and performance in smartphones," in *Workshop on Open Source and Design of Communication*, July 2011, pp. 1–6.

[149] Google, "Protocol Buffers - Google Developers," online https://developers.google.com/protocol-buffers/, Apr. 2012, retrieved: 03.2017.

[150] ——, "Protocol Buffers. Google's Data Interchange Format." online https://developers.google.com/protocol-buffers/, Apr. 2012, retrieved: 03.2017.

[151] B. Groza, S. Murvay, A. van Herrewege, and I. Verbauwhede, *Cryptology and Network Security*, ser. LNCS. Springer, Dec. 2012, vol. 7712, ch. LiBrA-CAN: A Lightweight Broadcast Authentication Protocol for Controller Area Networks, pp. 185–200.

[152] F. A. Hamza, "The USRP under 1.5X Magnifying Lens!" GNU Radio project, Tech. Rep., Jun 2008, rev. 1.0.

[153] J. L. Harbour, *Basics of Performance Measurement*. Quality Resources, April 1997.

[154] J. Harding, G. R. Powell, R. F. Yoon, J., C. Doyle, D. Sade, M. Lukuc, J. Simons, and J. Wang, "Vehicle-to-Vehicle Communications: Readiness of V2V Technology for Application," Washington, DC: National Highway Traffic Safety Administration, Tech. Rep. DOT HS 812 014, Aug. 2014.

[155] J. Härri, P. Cataldi, D. Krajzewicz, R. J. Blokpoel, Y. Lopez, and J. Leguay, "Modeling and Simulating ITS Applications with iTETRIS," in *6th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, Nov. 2011, pp. 33–40.

[156] H. Hartenstein and K. Laberteaux, Eds., *VANET Vehicular Applications and Inter-Networking Technologies*. Wiley, Oct. 2009.

[157] H. Hasbullah, I. A. Soomro, and J. A. Manan, "Denial of Service (DOS) Attack and Its Possible Solutions in VANET," *World Academy of Science, Engineering and Technology*, vol. 4, no. 348-352, May 2010.

[158] L. A. Hassnawi, R. B. Ahmad, M. . H. Salih, M. N. M. Warip, and M. Elshaikh, "Measurement Study on Packet Size and Packet Rate Effects over Vehicular ad hoc Network Performance," *Journal of Theoretical and Applied Information Technology*, vol. 70, no. 3, pp. 475 – 481, Dec. 2014.

[159] A. M. Hegland, E. Winjum, S. F. Mjolsnes, C. Rong, O. Kure, and P. Spilling, "A Survey of Key Management in Ad Hoc Networks," *IEEE Communications Surveys*, vol. 8, no. 3, pp. 48–66, Dec. 2006.

[160] R. Heinrichs, M. Fritzsche, and I. Radusch, "Improved Automotive Self Learning System using Hypothesis Test Triggered Forgetting to adapt to Change Points," in *IEEE Intelligent Vehicles Symposium*, June 2014, pp. 176–182.

[161] M. Heissenbüttel, T. Braun, T. Bernoulli, and M. Wälchli, "BLR: Beacon-Less Routing Algorithm for Mobile Ad-Hoc Networks," *Elsevier's Computer Communications Journal (Special Issue)*, vol. 27, no. 11, pp. 1076–1086, Feb. 2004.

[162] M. M. Hell and U. Kelling, "Power saving in CAN applications," in *13th international CAN Conference*, Mar. 2012, pp. 9–13.

[163] A. Hiller, C. Neumann, M. Mattheß, A. Festag, H. Santos, W. Zhang, C. Sorge, and M. Wiecker, "Deliverable D21.4, Spezifikation der Kommunikationsprotokolle," Sichere Intelligente Mobilität Testfeld Deutschland simTD, Tech. Rep., Sept. 2009.

[164] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle Trajectory Prediction based on Motion Model and Maneuver Recognition," in *IEEE International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 4363–4369.

[165] Y. Hu and K. P. Laberteaux, "Strong VANET Security on a Budget," in *In Proceedings of Workshop on Embedded Security in Cars*, 2006.

[166] J. Huang, L. L. Presti, B. Motella, and M. Pini, "GNSS spoofing detection: Theoretical analysis and performance of the Ratio Test metric in open sky," *ICT Express*, vol. 2, no. 1, pp. 37–40, Mar. 2016.

[167] L. Huang, K. Matruura, H. Yamane, and K. Sezaki, "Enhancing Wireless Location Privacy using Silent Period," in *IEEE Wireless Communications and Networking Conference*, 2005, pp. 1187–1192.

[168] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner, "Assessing the Spoofing Threat: Development of a Portable GPS Civilian Spoofer," in *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation*, Sept. 2008, pp. 2317–2325.

[169] *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 6: Wireless Access in Vehicular Environments*, IEEE Std., Rev. 802.11p-2010, July 2010, 802.11p-2010.

[170] *Intel Atom Processor Z5XX Series, Datasheet*, 3rd ed., Intel, June 2010, doc. No. 319535-003US.

[171] *2nd Generation Intel Core Processor Family, Datasheet, Vol.1*, 8th ed., Intel, June 2013, doc. No. 324641-008.

[172] Intel, "BERserk Vulnerability - Part 1: RSA signature forgery attack due to incorrect parsing of ASN.1 encoded DigestInfo in PKCS#1 v1.5," Intel Security: Advanced Threat Research, Tech. Rep., Sept. 2014.

[173] ——, "BERserk Vulnerability - Part 2: Certificate Forgery in Mozilla NSS," Intel Security: Advanced Threat Research, Tech. Rep., Oct. 2014.

[174] *IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services*, Intelligent Transportation Systems Commitee of the IEEE Vehicular Technology Society Std. P1609.3, Rev. 2010, Dec. 2010.

[175] *Draft Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages*, Intelligent Transportation Systems Commitee of the IEEE Vehicular Technology Society Std. P1609.2, Rev. D12, Jan. 2012.

[176] *IEEE Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages*, Intelligent Transportation Systems Commitee of the IEEE Vehicular Technology Society Std. P1609.2, Rev. 2013, Apr. 2013.

[177] ISO, "ISO/IEC 9945:2008 Information technology – Portable Operating System Interface (POSIX®)," May 2009.

[178] *ISO 11898-1:2003 Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signalling*, ISO Std. 43.040.15, Rev. 90.92 (2013-02-11), Feb. 2013.

[179] *ISO 17458-2:2013 Road vehicles – FlexRay communications system – Part 2: Data link layer specification*, ISO Std. 43.040.15, Rev. 60.60 (2013-01-21), Jan 2013.

[180] B. Iyidir and Y. Ozkazanc, "Jamming of GPS receivers," in *Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference*, Apr. 2004, pp. 747–750.

[181] A. Jafarnia-Jahromi, A. Broumandan, J. Nielsen, and G. Lachapelle, "GPS Vulnerability to Spoofing Threats and a Review of Antispoofing Techniques," *International Journal of Navigation and Observation*, vol. 9, pp. 1–16, July 2012.

[182] R. W. Johnson, J. L. Evan, P. Jacobsen, J. R. Thompson, and M. Christopher, "The Changing Automotive Environment: High-Temperature Electronics," *IEEE Transactions on Electronics Packaging Manufacturing*, vol. 27, no. 3, July 2004.

[183] M. T. Jones, "Kernel APIs, Part 3: Timers and lists in the 2.6 kernel," online: http://www.ibm.com/developerworks/library/l-timers-list/, Mar. 2010, retrieved: 03.2017.

[184] G. Kambourakis, E. Konstantinou, A. Douma, M. Anagnostopoulos, and G. Fotiadis, "Efficient Certification Path Discovery for MANET," *EURASIP Journal on Wireless Communications and Networking*, pp. 1–16, May 2010.

[185] F. Kargl, E. Schoch, B. Wiedersheim, and T. Leinmüller, "Secure and Efficient Beaconing for Vehicular Networks," in *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, 2008, pp. 82–83.

[186] K. Katsaros, R. Kernchen, M. Dianati, and D. Rieck, "Performance Study of a Green Light Optimized Spped Advisory (GLOSA) Application using an Integrated Cooperative ITS Simulation Platform," in *7th International Wireless Communications and Mobile Computing Conference*, July 2011, pp. 918–923.

[187] C. A. Kent and J. C. Mogul, "Fragmentation Considered Harmful," in *Proceedings of the ACM Workshop on Frontiers in Computer Communications Technology*, Aug. 1987, pp. 390–401.

[188] S. Khan and A. K. Pathan, Eds., *Wireless Networks and Security: Issues, Challenges and Research Trends*. Springer, 2013.

[189] H.-Y. Kim, D.-M. Kang, J.-H. Lee, and T.-M. Chung, "A Performance Evaluation of Cellular Network Suitability for VANET," *World Academy of Science, Engineering & Technology*, vol. 64, pp. 124–127, Apr. 2012.

[190] B. Kloiber, T. Strang, M. Röckl, and F. de Ponte-Muller, "Performance of CAM Based Safety Applications using ITS-G5A MAC in High Dense Scenarios," in *IEEE Intelligent Vehicles Symposium*, June 2011, pp. 654 – 660.

[191] B. Kloiber, T. Strang, F. de Ponte-Mueller, C. Rico Garcia, and M. Roeckl, "An Approach for Performance Analysis of ETSI ITS-G5A MAC for Safety Applications," in *The 10th International Conference on Intelligent Transport Systems Telecommunications*, Nov. 2010.

[192] M. Knezevic, V. Nikov, and P. Rombouts, "Low-Latency ECDSA Signature Verification - A Road Towards Safer Traffic -," *IACR Cryptology ePrint Archive*, pp. 862 – 877, Oct. 2014.

[193] J. Knudsen, *JAVA Cryptography*. O'Reilly, May 1998.

[194] N. Koblitz, "Elliptic Curve Cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, Jan. 1987.

[195] E. Koenders, D. Oort, and K. Rozema, "An open Local Dynamic Map," in *Proceedings 10th ITS European Congress*, June 2014.

[196] T. Kos, M. Grgic, and G. Sisul, "Mobile User Positioning in GSM/UMTS Cellular Networks," in *48th International Symposium Multimedia Signal Processing and Communications*, June 2006, pp. 185–188.

[197] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, and S. Checkoway, "Experimental Security Analysis of a Modern Automobile," in *31st IEEE Symposium on Security and Privacy*, May 2010, pp. 447–462.

[198] Kraftfahrt-Bundesamt, "Neuzulassungen von Personenkraftwagen im August 2014 nach Marken und Modellreihen," online: http://www.kba.de/DE/Statistik/Fahrzeuge/Neuzulassungen/MonatlicheNeuzulassungen/monatl_neuzulassungen_node.html, 2014, retrieved: 03.2017.

[199] H. Krishnan and A. Weimerskirch, "Verify-on-Demand - a pracitcal and scalable Approach for Broadcast Authentication in Vehicle-to-Vehicle Communication," *SAE International Journal of Passenger Cars - Mechanical Systems*, vol. 4, no. 1, pp. 536–546, Apr. 2011.

[200] T. Kürner and M. Schack, "Requirements and Challenges for the Development of Car-to-Car Channel Models," in *Car 2 Car Communication Consortium / COMeSafety: 1st open Workshop on Simulation*, Mar. 2007.

[201] R. Kyusakov, "Embeddable EXI Processor in C," http://exip.sourceforge.net/, Nov. 2014, retrieved: 03.2017.

[202] V. H. La and A. Cavalli, "Security Attacks and Solutions in Vehicular Ad-Hoc Networks: A Survey," *International Journal on AdHoc Networking Systems*, vol. 4, no. 2, pp. 1 – 20, Apr. 2014.

[203] L. Lamport, "Password Authentication with Insecure Communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, Nov. 1981.

[204] K. Lan and C. Chou, "Realistic mobility models for vehicular ad hoc network (vanet) simulations," in *8th International Conference on ITS Telecommunications*, Oct. 2008, pp. 362 – 366.

[205] S. Lefévre, J. Petit, R. Bajcsy, C. Laugier, and F. Kargl, "Impact of V2X Privacy Strategies on Intersection Collision Avoidance Systems," in *IEEE Vehicular Networking Conference*, Dec. 2013, pp. 71 – 78.

[206] T. Leinmüller, T., L. Buttyan, J. P. Hubaux, F. Kargl, R. Kroh, P. Papadimitratos, M. Raya, and E. Schoch, "SEVECOM - Secure Vehicle Communication," in *IST Mobile Summit*, Jun. 2006.

[207] Q. Li, T. Jinmei, and K. Shima, *IPv6 Core Protocols Implementation*. Morgan Kaufmann, Oct. 2006.

[208] Z. Li, C. Chigan, and C. Gao, "STCP2: Short-time Certificate-based Privacy Protection for Vehicular Ad Hoc Networks," in *IEEE Wireless Communications and Networking Conference*, Apr. 2013, pp. 1762–1767.

[209] X. Lin and R. Lu, *Vehicular Ad Hoc Network Security and Privacy*, S. Kartalopoulos, Ed. Wiley, 2015.

[210] B. Lipinski, W. Mazurczyk, K. Szczypiorski, and P. Smietanka, "Toward Effective Security Framework for Vehicular Ad-Hoc Networks," *Journal of Advances in Computer Networks*, vol. 3, no. 2, June 2015.

[211] T. Löffler and C. Kunze, "Test Car2X Applications," Vector Informatik GmbH, Tech. Rep., Dec. 2013.

[212] M. Lombardi, "The Accuracy and Stability of Quartz Watches," *Horological Journal*, pp. 57–59, Feb. 2008.

[213] M. A. Lombardi, "NIST Time and Frequency Services," NIST, NIST Special Publication 432, 2002.

[214] N. Lyamin, A. Vinel, M. Jonsson, and J. Loo, "Real-Time Detection of Denial-of-Service Attacks in IEEE 802.11p Vehicular Networks," *IEEE Communications Letters*, vol. 18, no. 1, pp. 110 – 113, Jan. 2014.

[215] M. Rondinone, et. al., "iTETRIS: A modular simulation platform for the large scale evaluation of cooperative ITS applications," *Simulation Modelling Practice and Theory*, vol. 34, pp. 99–125, May 2013.

[216] Z. Ma, F. Kargl, and M. Weber, "Pseudonym-On-Demand: A New Pseudonym Refill Strategy for Vehicular Communications," in *68th IEEE Vehicular Technology Conference*, Sept. 2008, pp. 1–5.

[217] A. Malhotra, I. Cohen, E. Brakke, and S. Goldberg, "Attacking the Network Time Protocol," Bosten University, Tech. Rep., 2015.

[218] A. M. Malla and R. K. Sahu, "Security Attacks with an Effective Solution for DOS Attacks in VANET," *International Journal of Computer Applications*, vol. 66, no. 22, pp. 45 – 49, Mar. 2013.

[219] J. J. Marciniak, *Encyclopedia of Software Engineering*, 2nd ed. John Wiley & Sons Inc, Jan. 2002.

[220] F. J. Martinez, C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni, "A Survey and Comparative Study of Simulators for Vehicular ad hoc Networks (VANETs)," *Wireless Communications & Mobile Computing*, vol. 11, no. 7, pp. 813–828, July 2011.

[221] M. Masdari and J. P. Barbin, "Distributed Certificate Management in Mobile Ad Hoc Networks," *International Journal of Applied Information Systems*, vol. 1, no. 1, pp. 33–40, Nov. 2012.

[222] M. N. Mejri, J. Ben-Othman, and M. Hamdi, "Survey on VANET Security Challenges and Possible Cryptographic Solutions," *Vehicular Communications*, vol. 1, pp. 53–66, Jan. 2014.

[223] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, 1st ed. CRC Press, Oct. 1996.

[224] C. Miller and C. Valasek, "Adventures in Automotive Networks and Control Units," Sept. 2013.

[225] V. S. Miller, "Use of Elliptic Curves in Cryptography," in *Advances in Cryptology - CRYPTO '85 Proceedings*, ser. LNCS, H. C. Williams, Ed., vol. 218. Springer Berlin Heidelberg, Dec. 2000, pp. 417–426.

[226] T. Mismar, "Cooperative Localization in Cellular Networks," Ph.D. dissertation, University of Toledo, May 2015.

[227] L. Molas, "Heap memory corruption in ASN.1 parsing code generated by Objective Systems Inc. ASN1C compiler for C/C++," online: https://github.com/programa-stic/security-advisories/tree/master/ObjSys/CVE-2016-5080, 2016 July, retrieved: 03.2017.

[228] P. Y. Montgomery, T. E. Humphreys, and B. M. Ledvina, "Receiver-Autonomous Spoofing Detection: Experimental Results of a Multi-antenna Receiver Defense Against a Portable Civil GPS Spoofer," in *Proceedings of the 2009 International Technical Meeting of The Institute of Navigation*, Jan. 2009, pp. 124 – 130.

[229] M. S. Morogan and S. Muftic, "Certificate Management in ad hoc Networks," in *Symposium on Applications and the Internet Workshops*, Jan. 2003, pp. 337–341.

[230] D. M'Raihi, M. Bellare, D. Naccache, and O. Ranen, "HOTP: An HMAC-Based One-Time Password Algorithm," Network Working Group, IETF, Tech. Rep. RFC: 4226, Dec. 2005.

[231] D. M'Raihi, S. Machani, M. Pei, and J. Rydell, "TOTP: Time-Based One-Time Password Algorithm," IETF, Tech. Rep. RFC:6238, May 2011.

[232] D. Mundy and D. Chadwick, "Comparing the Performance of Abstract Syntax Notation One (ASN.1) vs eXtensible Markup Language (XML)," in *Proceedings of the Terena Networking Conference*, May 2003, pp. 182–196.

[233] P. Nisch, "Security Issues in Modern Automotive Systems," June 2012.

[234] NIST, *Announcing the Advanced Encryption Standard (AES)*, Information Technology Laboratory, National Institute of Standards and Technology Std., Rev. FIPS PUB 197, Nov. 2001.

[235] ——, *Secure Hash Standard (SHS)*, Information Technology Laboratory, National Institute of Standards and Technology Std., Rev. FIPS PUB 180-4, Mar. 2012.

[236] NIST, "WWVB Coverage Area," online: http://www.nist.gov/pml/div688/grp40/vb-coverage.cfm, Dec. 2015, retrieved: 03.2017.

[237] N. Nowdehi and T. Olovsson, "Experiences from Implementing the ETSI ITS Secured Message Service," in *IEEE Intelligent Vehicles Symposium*, 2014, pp. 1055–1060.

[238] NXP, "PCA2129, Automotive accurate RTC with integrated quartz crystal," online: https://cache.nxp.com/documents/data_sheet/PCA2129.pdf, Dez. 2014, 5th edition, retrieved: 03.2017.

[239] M. Nystrom and B. Kaliski, "PKCS #10: Certification Request Syntax Specification, Version 1.7," Network Working Group, IETF, Tech. Rep. RFC2986, Nov. 2000.

[240] OpenSSL Software Foundation, "OpenSSL: Cryptography and SSL/TLS Toolkit," https://www.openssl.org/, May 2016, retrieved: 03.2017.

[241] OpenStreetMap Foundation, "OpenSteetMap," online: http://www.openstreetmap.org, Oct. 2014, retrieved: 03.2017.

[242] OSS Nokalva, Inc, "Alternative Binary Representations of the XML Information Set based on ASN.1," online: www.w3.org/2003/08/binary-interchange-workshop/32-OSS-Nokalva-Position-Paper-updated.pdf, Aug. 2013, retrieved: 03.2017.

[243] ——, "ASN.1 Tools for C - Overview," online: http://www.oss.com/asn1/products/asn1-c/asn1-c.html, Jan. 2014, retrieved: 03.2017.

[244] C. Paar and J. Pelzl, *Understanding Cryptography*, 2nd ed. Springer, 2010.

[245] M. Paik, "Stragglers of the Herd get eaten: Security Concerns for GSM Mobile Banking Applications," in *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*, Feb. 2010, pp. 54 – 59.

[246] G. Paoloni, "How to Benchmark Code Execution Times on Intel IA-32 and IA-64 Instruction Set Architectures," Intel, White Paper 324264-001, Sept. 2010.

[247] M. Peden, R. Scurfield, D. Sleet, D. Mohan, A. A. Hyder, E. Jarawan, and C. Mathers, "World Report on Road Traffic Injury Prevention," World Health Organization, Geneva, Tech. Rep., 2004.

[248] D. Peintner, H. Kosch, and J. Heuer, "Efficient XML Interchange for Rich Internet Applications," in *IEEE International Conference on Multimedia and Expo*, June 2009, pp. 149–152.

[249] S. K. Pell and C. Soghoian, "Your Secret Stingray's No Secret Anymore: The Vanishing Government Monopoly over Cell Phone Surveillance and Its Impact on National Security and Consumer Privacy," *Harvard Journal of Law and Technology*, vol. 28, no. 1, pp. 1–76, Dec. 2014.

[250] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: Security Protocols for Sensor Networks," *Wireless Networks*, vol. 8, pp. 521–534, Sept. 2002.

[251] J. Petit, F. Schaub, M. Feiri, and F. Kargl, "Pseudonym Schemes in Vehicular Networks: A Survey," *IEEE Communication Surveys & Tutorials*, vol. 17, no. 1, pp. 228 – 255, Mar. 2015.

[252] Physikalisch-Technische Bundesanstalt (PTB), "Dissemination of legal time," online: http://www.ptb.de/cms/en/fachabteilungen/abt4/fb-44/ag-442/dissemination-of-legal-time.html, Aug. 2011, retrieved: 03.2017.

[253] D. Piester, A. Bauch, J. Becker, and A. Hoppmann, "Time and Frequency Broadcast with DCF77," in *43rd Annual Time and Time Interval (PTTI) Systems and Applications Meeting*, Nov. 2011, pp. 185–196.

[254] A. A. Pirzada and C. McDonald, "Kerberos Assisted Authentication in Mobile Ad-hoc Networs," in *27th Australasian Computer Science Conference*, 2004, pp. 41–46.

[255] M. L. Psiaki, T. E. Humphreys, and B. Stauffer, "Attackers can Spoof Navigation Signals without our Knowledge. Here's how to Fight Back GPS Lies," *IEEE Spectrum*, vol. 53, no. 8, pp. 26 – 53, Aug. 2016.

[256] M. L. Psiaki, S. P. Powell, and W. O'Hanlon, "GNSS Spoofing Detection - Correlating Carrier Phase with Rapid Antenna Motion," *GPS World*, pp. 53 – 58, June 2013.

[257] R. M. Stallman and the GCC Developer Community, *Using the GNU Compiler Collection, For GCC version 4.8.2*, Free Software Foundation, Oct. 2013.

[258] K. B. Rasmussen, S. Capkun, and M. Cagalj, "SecNav: Secure Broadcast Localization and Time Synchronization in Wireless Networks," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, Sep. 2007, pp. 310–313.

[259] G. Remy, S.-M. Senouci, F. Jan, and Y. Gourhant, "LTE4V2X: LTE for a Centralized VANET Organization," in *IEEE Global Telecommunications Conference*, Dec. 2011, pp. 1–6.

[260] K. Ren, W. Lou, K. Zeng, and P. J. Moran, "On Broadcast Authentication in Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 11, pp. 4136–4144, Nov. 2007.

[261] W. B. Ribbens, N. P. Mansour, G. Luecke, C. W. Battle, W. C. Jones, and L. E. Mansir, Eds., *Understanding Automotive Electronics*, 6th ed. Newnes, Jan. 2003.

[262] R. Riebl, H. J. Gunther, C. Facchi, and L. Wolf, "Artery: Extending Veins for VANET Applications," in *4th International Conference on Models and Technologies for Intelligent Transport Systems*, June 2015, pp. 450–456.

[263] G. F. Riley and T. R. Henderson, "The ns-3 Network Simulator," in *Modeling and Tools for Network Simulation*, K. Wehrle, M. Günes, and J. Gross, Eds. Springer Berlin Heidelberg, 2010, pp. 15–34.

[264] D. Robil and A. Vinel, Eds., *Roadside Networks for Vehicular Communications: Architectures, Applications and Test Fields*. Information Science Reference, Oct. 2012.

[265] S. Robinson, *Simulation - The Practice of Model Development and Use*. Wiley, Dec. 2003.

[266] K. Roscher, S. Bittl, A. A. Gonzalez, M. Myrtus, and J. Jiru, "ezCar2X: Rapid-Prototyping of Communication Technologies and Cooperative ITS Applications on Real Targets and Inside Simulation Environments," in *11th Conference Wireless Communication and Information*. vwh, Oct. 2014, pp. 51 – 62.

[267] K. Roscher, J. Jiru, A. Gonzalez, and W. Heidrich, "ezCar2X: A Modular Software Framework for Rapid Prototyping of C2X Applications," in *9th ITS European Congress*, June 2013.

[268] M. Rossberg, R. Golembewski, and G. Schaefer, "Attack-Resistant Distributed Time Synchronization for Virtual Private Networks," in *21st International Conference on Computer Communications and Networks*, Aug. 2012, pp. 1–8.

[269] S. Röttger, "Analysis of the NTP autokey Procedures," Master Thesis, Technical University Braunschweig, Feb. 2012.

[270] A. Rügamer and D. Kowalewski, "Jamming and Spoofing of GNSS Signals - An Underestimated Risk?!" in *FIG Working Week*, Mar. 2015, pp. 1–21.

[271] *Dedicated Short Range Communications (DSRC) Message Set Dictionary*, SAE Std. J2735_201 603, Mar. 2016.

[272] M. Saini, A. Alelaiwi, and A. El Saddik, "How Close are We to Realizing a Pragmatic VANET solution? A Meta-Survey," *ACM Computing Surveys*, vol. 48, no. 2, pp. 29:1–29:40, Nov. 2015.

[273] F. Scheuer, K. Plößl, and H. Federrath, "Preventing Profile Generation in Vehicular Networks," in *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Oct. 2008, pp. 520–525.

[274] E. Schoch, T. Gansen, F. Kargl, N. Bißmayer, and B. Lonc, "C2C-CC Privacy Memo," 2012, revision 0.5.

[275] E. Schoch and F. Kargl, "On the Efficientcy of Secure Beaconing in VANETs," in *Proceedings of the Third ACM Conference on Wireless Network Security*, Mar. 2010, pp. 111 – 116.

[276] E. Schoch, F. Kargl, T. Leinmüller, S. Schlott, and P. Papadimitratos, *Security and Privacy in Ad-Hoc and Sensor Networks, Third European Workshop, ESAS 2006, Revised Selected Papers*. Springer, 2006, vol. 4357, ch. Impact of Pseudonym Changes on Geographic Routing in VANETs, pp. 43–57.

[277] B. Schuenemann, "V2X Simulation Runtime Infrastructure VSimRTI: An Assessment Tool to Design Smart Traffic Management Systems," *Computer Networks*, vol. 55, no. 14, pp. 3189–3198, Oct. 2011.

[278] H. Schumacher, M. Schack, and T. Kürner, "Coupling of Simulators for the Investigation of Car-to-X Communication Aspects," in *IEEE Asia-Pacific Services Computing Conference*, Dec. 2009, pp. 58–63.

[279] T. Schütze, "Automotive Security: Cryptography for Car2X Communication," in *Embedded World Conference*, Mar. 2011, pp. 1–16.

[280] H. Schweppe and Y. Roudier, "Security Issues in Vehicular Systems : Threats, Emerging Solutions and Standards," in *5th Conference on Network Architectures and Information Systems Security*, May 2010, pp. 1–5.

[281] H. Schweppe, Y. Roudier, B. Weyl, and L. Apvrille, "Car2X Communication: Securing the Last Meter - A Cost-Effective Approach for Ensuring Trust in Car2X Applications Using In-Vehicle Symmetric Cryptography," in *IEEE Vehicular Technology Conferenece - Fall*, Sept. 2011, pp. 1–5.

[282] J. Sen, M. G. Chandra, P. Balamuradlidhar, and S. G. Harihara, "A Scheme of Certificate Authority for Ad Hoc Networks," in *18th International Workshop on Database and Expert Systems Applications*, Sept. 2007, pp. 615–619.

[283] A. Serjantov and G. Danezis, "Towards an Information Theoretic Metric for Anonymity," in *Privacy Enhancing Technologies*, ser. LNCS, vol. 2482, June 2003, pp. 41–53.

[284] J. Seward, N. Nethercote, J. Weidendorfer, and V. D. Team, *Valgrind 3.3*, 1st ed. Network Theory Ltd., May 2008.

[285] D. Seydel, S. Bittl, J. Pfeiffer, J. Jiru, H. Beckmann, K. Frankl, and B. Eissfeller, "An Evaluation Methodology for VANET Applications combining Simulation and Multi-sensor Experiments," in *2nd International Conference on Vehicular Intelligent Transport Systems*, Apr. 2016, pp. 213–224.

[286] A. Shalk, R. Borgaonkar, N. Asokan, V. Nleml, and J.-P. Selfert, "Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems," in *arXiv.org*, Nov. 2015. [Online]. Available: http://arxiv.org/abs/1510.07563

[287] C. Sommer and F. Dressler, *Vehiclular Networking*. Cambridge University Press, 2015.

[288] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, Jan. 2011.

[289] B. Sosinsky, *Networking Bible*. Wiley, Sept. 2009.

[290] R. Stahlmann, A. Festag, A. Tomatis, I. Radusch, and F. Fischer, "Starting European Field Tests for Car-2-X Communication: The Drive C2X Framework," in *Proceedings of 18th ITS Worls Congress and Exhibition*, Oct. 2011, pp. 1 – 9.

[291] H. Stübing, *Multilayered Security and Privacy Protection in Car-to-X Networks*, 1st ed. Springer Vieweg, 2013.

[292] I. A. Sumra, I. Ahmad, H. Hasbullah, and J. bin Ab Manan, "Classes of Attacks in VANET," in *Saudi International Electronics, Communications and Photonics Conference*, Apr. 2011, pp. 1–5.

[293] Y. Sun, X. Lin, R. Lu, X. Shen, and J. Su, "Roadside Units Deployment for Efficient Short-Time Certificate Updating in VANETs," in *IEEE International Conference on Communications*, May 2010, pp. 1–5.

[294] Y. Sun, B. Zhang, B. Zhao, X. Su, and J. Su, "Mix-zones Optimal Deployment for Protecting Location Privacy in VANET," *Peer-to-Peer Networking and Applications*, vol. 8, no. 6, pp. 1108–1121, 2015.

[295] M. Sybis and A. Langowski, "Throughput Enhancement for the 802.11p Communication," in *IEEE Vehicular Networking Conference*, Dec. 2014, pp. 97 – 100.

[296] Symmetricom, "Timing and Synchronization for LTE-TDD and LTE-Advanced Mobile Networks," Symmetricom, Inc.," White Paper, 2013, https://www.aventasinc.com/whitepapers/WP-Timing-Sync-LTE-SEC.pdf, retrieved: 03.2017.

[297] K. Teichel, D. Sibold, and S. Milius, "First Results of a Formal Analysis of the Network Time Security Specification," in *Security Standardisation Research*, ser. LNCS, vol. 9497, Dec. 2015, pp. 218–245.

[298] The University of Texas in Austin, "UT Austin Researchers Spoof Superyacht at Sea," online: http://www.engr.utexas.edu/features/superyacht-gps-spoofing, July 2013, retrieved: 03.2017.

[299] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the Requirements for Successful GPS Soppofing Attacks," in *Proceedings of the 18th ACM conference on Computer and Communications Security*, Oct. 2011, pp. 75–86.

[300] N. O. Tippenhauer, K. B. Rasmussen, C. Pöpper, and S. Capkun, "Attacks on Public WLAN-based Positioning," in *Proceedings of the 7th International Conference on Mobile Systems, Applications and Services*, June 2009, pp. 29–40.

[301] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed. The Institution of Electrical Engineers, Mar. 2005.

[302] A. Tomandl, D. Herrmann, K.-P. Fuchs, H. Federrath, and F. Scheuer, "VANETsim: An open source simulator for security and privacy concepts in VANETs," in *International Conference on High Performance Computing Simulation*, July 2014, pp. 543 – 550.

[303] A. Tomandl, F. Scheuer, and H. Federrath, "Simulation-based Evaluation of Techniques for Privacy Protection in VANETs," in *IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications*, Oct. 2012, pp. 165–172.

[304] O. K. Tonguz, W. Viriyasitavat, and F. Bai, "Modeling Urban Traffic: A Cellular Automata Approach," in *IEEE Communications Magazin*, May 2009, pp. 142 – 150.

[305] T. Toroyan, "Global Status Report on Road Safety," World Health Organisation, Tech. Rep., 2013.

[306] u-blox, "NEO-M8 series," https://www.u-blox.com/en/product/neo-m8-series, 2016, retrieved: 03.2017.

[307] R. Uzcategui and G. Acosta-Marum, "WAVE: A Tutorial," *IEEE Communications Magazine*, vol. 47, no. 5, pp. 126–133, May 2009.

[308] A. Van Herrewege, D. Singelée, and I. Verbauwhede, "CANAuth - A Simple, Backward Compatible Broadcast Authentication Protocol for CAN bus," in *ECRYPT Workshop on Lightweight Cryptography 2011*, Jan. 2011.

[309] A. Varga, *Modeling and Tools for Network Simulation*. Springer, 2010, ch. OMNeT++, pp. 35–59.

[310] Vector Informatik GmbH, "Development & Test Tool CANoe.Car2x," online: http://vector.com/vi_canoe_car2x_en.html, 2016, retrieve: 07.2016.

[311] *Efficient XML Interchange (EXI) Format*, W3C Std. 1.0, Rev. 2nd, Feb. 2014.

[312] I. Wagner and D. Eckhoff, "Privacy Assessment in Vehicular Networks using Simulation," in *Proceedings of the Winter Simulation Conference*, Dec. 2014, pp. 3155–3166.

[313] T. Wambach and K. Bräunlich, "Retrospective Study of Third-party Web Tracking," in *2nd International Conference on Information Systems Security and Privacy*, Feb. 2016, pp. 138–145.

[314] C.-X. Wang, X. Cheng, and D. I. Laurenson, "Vehicle-to-Vehicle Channel Modeling and Measurements: Recent Advances and Future Challenges," *IEEE Communications Magazine*, vol. 4, no. 11, pp. 96–103, Nov. 2009.

[315] K. Wang, S. Chen, and A. Pan, "Time and Position Spoofing with Open Source Projects," in *black hat Europe*, Nov. 2015.

[316] S.-W. Wang, "Allocation of Roadside Units for Certificate Update in Vehicular Ad Hoc Network Environments," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 3, pp. 114–121, 2015.

[317] S.-W. Wang and M.-Y. Chang, "Roadside Units Allocation Algorithms for Certificate Update in VANET Environments," in *17th Asia Pacific Conference on Communications*, Oct. 2011, pp. 472–477.

[318] S.-Y. Wang, C.-L. Chou, and C.-M. Yang, "EstiNet Openflow Network Simulator and Emulator," *IEEE Communications Magazine*, vol. 51, no. 9, pp. 110 – 117, Sept. 2013.

[319] B. Watson, "Network Protocol Design with Machiavellian Robustness," Ph.D. dissertation, Macquarie University, Faculty of Science, Department of Computing, Nov. 2010.

[320] A. Weimerskirch, "V2X Security & Privacy: The Current State and Its Future," in *Proceedings 18th ITS World Congress*, Oct. 2011.

[321] K. D. Wesson, "GPS Spoofing & Implications for Telecom," Sprint Synchronization Conference, Sept. 2013.

[322] ——, "Secure Navigation and Timing Without Local Storage of Secret Keys," Ph.D. dissertation, University of Texas at Austin, Faculty of the Graduate School, May 2014.

[323] E. Whelan, "SNMP and Potential ASN.1 Vulnerabilities," CISSP, Tech. Rep., Dec. 2002.

[324] W. Whyte, A. Weimerskirch, A. Kumar, and T. Hehn, "A Security Credential Management System for V2V Communications," in *IEEE Vehicular Networking Conference*, Dec. 2013, pp. 1–8.

[325] B. Wiedersheim, Z. Ma, F. Kargl, and P. Papadimitratos, "Privacy in Inter-Vehicular Networks: Why simple pseudonym change is not enough," in *Seventh International Conference on Wireless On-demand Network Systems and Services*, Feb. 2010, pp. 176–183.

[326] Wikipedia, "NITZ," online: https://en.wikipedia.org/wiki/NITZ, June 2016, retrieved: 03.2017.

[327] M. Wolf, "Security Engineering for Vehicular IT Systems," Ph.D. dissertation, Ruhr-Universität Bochum, 2008.

[328] M. Wolf, A. Weimerskirch, and C. Paar, "Security in Automotive Bus Systems," in *Proceedings of the Workshop on Embedded Security in Cars*, Nov. 2004.

[329] H. Wymeersch, J. Lien, and W. M. Z., "Cooperative Localization in Wireless Networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.

[330] D. Xiao, X. Liao, G. Tang, and C. Li, "Using Chebyshev Chaotic Map to Construct Infinite Length Hash Chains," in *International Conference on Communications, Circuits and Systems*, vol. 1, June 2004, pp. 11–12.

[331] G. Yan, S. Olariu, and M. C. Weigle, "Providing VANET Security through Active Position Detection," *Computer Communications, Mobility Protocols for ITS/VANET*, vol. 31, no. 12, pp. 2883–2897, July 2008.

[332] P. K. Yuen, *Practical Cryptology and Web Security*. Addison-Wesley Educational Publishers, Oct. 2005.

[333] A. T. Zamani and S. Zubair, "Key Management Scheme in Mobile Ad Hoc Networks," *International Journal of Emerging Research in Management & Technology*, vol. 3, no. 4, pp. 157–165, Apr. 2014.

[334] Q. Zeng, H. Li, and L. Qian, "GPS Spoofing Attack on Time Synchronization in Wireless Networks and Detection Scheme Design," in *IEEE Military Communications Conference*, Oct. 2012, pp. 1–5.

[335] Z. Zhang, S. Gong, A. D. Dimitrovski, and H. Li, "Time Synchronization Attack in Smart Grid: Impact and Analysis," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 87–98, Jan. 2013.

[336] Z. Zhang, H. Trinkle, M. Li, and A. D. Dimitrovski, "Combating Time Synchronization Attack: A Cross Layer Defense Mechanism," in *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems*, Apr. 2013, pp. 141–149.

[337] L. Zimmermann, A. Goetz, G. Fischer, and R. Weigel, "GSM Mobile Phone Localization using Time Difference of Arrival and Angle of Arrival Estimation," in *9th International Multi-Conference on Systems, Signals and Devices*, Mar. 2012, pp. 1–7.